МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
имени М.В. ЛОМОНОСОВА

ОЛИМПИАДНАЯ РАБОТА

Наименование олимпиады школьников: **«Ломоносов»**

Профиль олимпиады: **Информатика**

ФИО участника олимпиады: **Константиов Юрий Максимович**

Класс: **11 класс**

Технический балл: **85**

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	67	100	36	72	0

Дополнительный балл: 55

Задание 1. Попытка 1.

```
def get_factor(k):
	res = 1
	for i in range(1, k + 1):
		res *= i
	return res


def get_num(c):
	if (ord(c) >= ord('0') and ord(c) <= ord('9')):
		return 0 + ord(c) - ord('0')
	if (ord(c) >= ord('a') and ord(c) <= ord('z')):
		return 10 + ord(c) - ord('a')
	if (ord(c) >= ord('A') and ord(c) <= ord('Z')):
		return 36 + ord(c) - ord('A')


def get_val(s):
	n = len(s)
	power = n
	res = 0
	for i in range(n):
		res += get_factor(power) * get_num(s[i])
		power -= 1
	return res


#
```

```
# print(get_factor(0))
# print(get_val('8'))
#
#
# print(ord('A'))
# print(get_num('B'))
# print(get_val("4020"))



k = get_factor(int(input()))
mas = []
num = int(input())
for i in range(num):
        mas.append(input())
# print(mas)
max_n = -1
max_s = ""
for i in range(len(mas)):
        val = get_val(mas[i])
        if (val % k == 0 and val > max_n):
                max_n = val
                max_s = mas[i]
if (max_n == -1):
        print(-1)
        exit(0)
```

```
is_ahh = False
max_s_fixed = ""
for i in range(len(max_s)):
        if (is_ahh == True):
                max_s_fixed += max_s[i]
                continue
        if (is_ahh == False and (max_s[i] != '0' or i == len(max_s) - 1)):
                max_s_fixed += max_s[i]
                is_ahh = True




print(max_s_fixed)
ans = []
for i in range(len(mas)):
        if (get_val(mas[i]) == get_val(max_s)):
                print(i + 1)
```

Задание 2. Попытка 1.

```
def get_factor(k):
    res = 1
    for i in range(1, k + 1):
        res *= i
    return res


def get_num(c):
    if (ord(c) >= ord('0') and ord(c) <= ord('9')):
        return 0 + ord(c) - ord('0')
    if (ord(c) >= ord('a') and ord(c) <= ord('z')):
        return 10 + ord(c) - ord('a')
    if (ord(c) >= ord('A') and ord(c) <= ord('Z')):
        return 36 + ord(c) - ord('A')
    return -1


def get_val(s):
    n = len(s)
    power = n
    res = 0
    for i in range(n):
        res += get_factor(power) * get_num(s[i])
        power -= 1
    return res
```

```
#
# print(get_factor(0))
# print(get_val('8'))
#
#
# print(ord('A'))
# print(get_num('B'))
# print(get_val("4020"))


mas = [1, 4, 2, 3]
mas.pop(2)


n = int(input())
s = input()
mas = []
for i in range(n):
        if (get_num(s[i]) == -1):
                continue
        mas.append(s[i])
finded = False
s = ""
for i in range(61):
        max_ind = -1
        max_value = 100000000
        ind = i + 1
```

```
        find_in_this_turn = False
        t = ''
        if (len(mas) == 0):
            break

        for j in range(len(mas)):
            if (get_num(mas[j]) <= ind and get_num(mas[j]) < max_value):
                max_value = get_num(mas[j])
                t = mas[j]
                find_in_this_turn = True
                max_ind = j
        if (find_in_this_turn == True and max_ind != -1):
            s += t

            mas.pop(max_ind)
            finded = True
        else:
            break

if (finded == False):
        print(-1)
        exit(0)



n = len(s)
```

```
for i in range(n - 1, -1, -1):
        ind = i + 1
        max_value = -1
        max_ind = -1
        t = ''
        for j in range(len(mas)):
                if (get_num(mas[j]) > max_value and get_num(mas[j]) <= ind):
                        max_value = get_num(mas[j])
                        t = mas[j]
                        max_ind = j
        if (max_ind == -1):
                continue
        else:
                mas.pop(max_ind)
                mas.append(s[i])
                s = s[0:i] + t + s[i + 1:]
sans = ""
for i in range(len(s) - 1, -1, -1):
        sans += s[i]



ans_fixed = ""


finded = False
```

```
for i in range(0, len(sans)):
        if (finded == True):
                ans_fixed += sans[i]
                continue
        if (finded == False and (get_num(sans[i]) != 0 or i == (len(sans) - 1))):
                ans_fixed += sans[i]
                finded = True



print(ans_fixed)
```

Задание 3. Попытка 1.

```
/*\YOURiSST_xD\*/
#include <iostream>
#include <string>
#include <vector>
#include <set>
#include <queue>
#include <deque>
#include <cmath>
#include <algorithm>
#include <numeric>
#include <map>
#include <chrono>
#include <random>
#include <bitset>
#include <unordered_set>
#include <unordered_map>
#include <cassert>

#define sz(a) ((int)((a).size()))
#define get(_type, _x) \
    _type _x; \
    cin >> _x
#define mp(a, b) make_pair(a, b)
#define e "\n"
```

```
#define pb push_back
#define eb emplace_back
#define ft first
#define sc second
#define vi vector<int>
#define vl vector<long long>
#define vvi vector<vector<int>>
#define vvl vector<vector<long long>>
#define pii pair<int, int>
#define pll pair<long long, long long>
#define all(a) (a).begin(), (a).end()
#define rall(a) (a).rbegin(), (a).rend()
#define max max<int>
#define min min<int>
#define Time (double)clock() / CLOCKS_PER_SEC
#define filein(FILE) freopen(FILE, "r", stdin)
#define fileout(FILE) freopen(FILE, "w", stdout)
typedef long long ll;
typedef long double ld;
typedef unsigned long long ull;
using namespace std;
mt19937 rng(chrono::steady_clock::now().time_since_epoch().count());
mt19937_64 rng_ll(chrono::steady_clock::now().time_since_epoch().count());

template<typename T1, typename T2>
```

```
ostream &operator<<(ostream &out, pair<T1, T2> p) {
    out << '<' << p.first << " | " << p.second << '>';
    return out;
}

template<typename T>
ostream &operator<<(ostream &out, const vector<T> &vec) {
    for (int i = 0; i < vec.size(); ++i) {
        out << vec[i];
        if (i + 1 != vec.size()) {
            out << " ";
        }
    }
    return out;
}

template<typename T>
ostream &operator<<(ostream &out, const vector<vector<T>> &vec) {
    for (int i = 0; i < vec.size(); ++i) {
        for (int j = 0; j < vec[i].size(); ++j) {
            out << vec[i][j];
            if (j + 1 != vec[j].size()) {
                out << " ";
            }
        }
```

```
        if (i != vec.size() - 1)
            out << e;
    }
    return out;
}



template<typename T>
void _dbg(const char *_s, T _h) { cerr << _s << " = " << _h << "\n"; }

template<typename T, typename... Ts>
void _dbg(const char *_s, T _h, Ts... _t) {
    for (int _b = 0; ((_b += *_s == '(') -= *_s == ')') != 0 || *_s != ',';)cerr << *_s++;
    cerr << " = " << _h << ",";
    _dbg(_s + 1, _t...);
}

#ifdef LOCAL
#define dbg(...) _dbg(#__VA_ARGS__, __VA_ARGS__)
#else
#define dbg(...)
#endif



int binpow(int a, int n) {
```

```
    int result = 1;
    while (n) {
        if (n & 1)
            result *= a;
        a *= a;
        n >>= 1;
    }
    return result;
}

/*
const int ML = 240;
char memory[ML << 20];
size_t ptr_ = 0;

void* operator new(size_t how_much) {
    ptr_ += how_much;
    assert(ptr_ < ML * 1024 * 1024);
    return memory + ptr_ - how_much;
}

void operator delete(void*) { }
*/
```

```
#define int int
int const INF = 2 * 1e9;
ll const LINF = 1e18;
const int MAXN = 4 * 100 * 1001; // 1e5 + 0
//ll const CONST = 9223372036854775807;
long double const PI = 3.14159265358979323846;
//int const MAXN = 101;
const int SQRT = 350;
const int LOG = 21;
const int MOD = 1e9 + 7;

vvi g;
int ssz = 0; int n;
set<vector<pii>> omegalul;
int cnt = 0;
void search_left(int x, int y, vector<pii> path, vvi used, int counter) {
    used[x][y] = 1;
    used[y][x] = 1;
    dbg(x, y);
    counter++;
    if (counter == ssz) {
        cnt++;
        vector<pii> path_2 = path, path_3 = path;
        if (y + 1 < n && !used[x][y + 1]) {
```

```
        path_2.eb(x, y + 1);
        sort(all(path_2));
        omegalul.emplace(path_2);
    }
    if (y - 1 > -1 && !used[x][y - 1]) {
        path_3.eb(x, y - 1);
        sort(all(path_3));
        omegalul.emplace(path_3);
    }
    sort(all(path));
    omegalul.emplace(path);
    return;
}
int x1 = y - 1;
int x2 = y + 1;
if (x1 > -1) {
    if (!used[x1][y]) {
        used[x1][y] = 1;
        used[y][x1] = 1;
        vector<pii> new_path1 = path;
        new_path1.eb(x1, y);
        if (x1 + 1 < n && g[x1][x1 + 1] && !used[x1][x1 + 1]) {
            used[x1][x1 + 1] = 1;
            used[x1 + 1][x1] = 1;
```

```
            search_left(x1, x1 + 1, new_path1, used, counter);
        }
        if (g[x1][x1] && !used[x1][x1]) {
            used[x1][x1] = 1;
            search_left(x1, x1, new_path1, used, counter);
        }
        if (x1 - 1 > -1 && g[x1][x1 - 1] && !used[x1][x1 - 1]) {
            used[x1][x1 - 1] = 1;
            used[x1 - 1][x1] = 1;
            search_left(x1, x1 - 1, new_path1, used, counter);
        }
        /*used.clear();
        path.clear();*/
    }
}
if (x2 < n) {
    x1 = x2;
    if (!used[x1][y]) {
        used[x1][y] = 1;
        used[y][x1] = 1;
        vector<pii> new_path1 = path;
        new_path1.eb(x1, y);
        if (x1 + 1 < n && g[x1][x1 + 1] && !used[x1][x1 + 1]) {
            used[x1][x1 + 1] = 1;
            used[x1 + 1][x1] = 1;
```

```
            search_left(x1, x1 + 1, new_path1, used, counter);
        }
        if (g[x1][x1] && !used[x1][x1]) {
            used[x1][x1] = 1;
            search_left(x1, x1, new_path1, used, counter);
        }
        if (x1 - 1 > -1 && g[x1][x1 - 1] && !used[x1][x1 - 1]) {
            used[x1][x1 - 1] = 1;
            used[x1 - 1][x1] = 1;
            search_left(x1, x1 - 1, new_path1, used, counter);
        }
        /*used.clear();
        path.clear();*/
      }
   }
}

void solve() {
   cin >> n;
   g.resize(n, vi(n));
   string t; cin >> t;
   string t2;
   vector<pii> edges;
   vector<pii> edges_rev;
   while (true) {
```

```
    cin >> t;
    if (t == "END") {
        break;
    }
    cin >> t2;
    int x = stoi(t);
    int y = stoi(t2);
    dbg(x, y);
    g[x - 1][y - 1] = 1;
    g[y - 1][x - 1] = 1;
    edges.eb(x - 1, y - 1);
    edges_rev.eb(y - 1, x - 1);
}
ssz = sz(edges);
vvi used(n, vi(n, 0));

for (auto & [x, y] : edges) {
    vector<pii> path;
    search_left(x, y, path, used, 0);
    if (x + 1 < n) {
        vector<pii> new_path1 = {{x, x + 1}};
        search_left(x, y, new_path1, used, 0);
    }
    if (x - 1 > -1) {
        vector<pii> new_path2 = {{x, x - 1}};
```

```
            search_left(x, y, new_path2, used, 0);
        }
    }
    dbg(sz(omegalul), cnt);
    for (auto & i : omegalul) {

        cout << i << e << e;
    }
    cout << sz(omegalul);
}



signed main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    cout.tie(nullptr);
    solve();
    dbg(Time);
    return 0;
}
```

Задание 3. Попытка 2.

```
/*\YOURiSST_xD\*/
#include <iostream>
#include <string>
#include <vector>
#include <set>
#include <queue>
#include <deque>
#include <cmath>
#include <algorithm>
#include <numeric>
#include <map>
#include <chrono>
#include <random>
#include <bitset>
#include <unordered_set>
#include <unordered_map>
#include <cassert>

#define sz(a) ((int)((a).size()))
#define get(_type, _x) \
    _type _x; \
    cin >> _x
#define mp(a, b) make_pair(a, b)
#define e "\n"
```

```
#define pb push_back
#define eb emplace_back
#define ft first
#define sc second
#define vi vector<int>
#define vl vector<long long>
#define vvi vector<vector<int>>
#define vvl vector<vector<long long>>
#define pii pair<int, int>
#define pll pair<long long, long long>
#define all(a) (a).begin(), (a).end()
#define rall(a) (a).rbegin(), (a).rend()
#define max max<int>
#define min min<int>
#define Time (double)clock() / CLOCKS_PER_SEC
#define filein(FILE) freopen(FILE, "r", stdin)
#define fileout(FILE) freopen(FILE, "w", stdout)
typedef long long ll;
typedef long double ld;
typedef unsigned long long ull;
using namespace std;
mt19937 rng(chrono::steady_clock::now().time_since_epoch().count());
mt19937_64 rng_ll(chrono::steady_clock::now().time_since_epoch().count());

template<typename T1, typename T2>
```

```
ostream &operator<<(ostream &out, pair<T1, T2> p) {
    out << '<' << p.first << " | " << p.second << '>';
    return out;
}

template<typename T>
ostream &operator<<(ostream &out, const vector<T> &vec) {
    for (int i = 0; i < vec.size(); ++i) {
        out << vec[i];
        if (i + 1 != vec.size()) {
            out << " ";
        }
    }
    return out;
}

template<typename T>
ostream &operator<<(ostream &out, const vector<vector<T>> &vec) {
    for (int i = 0; i < vec.size(); ++i) {
        for (int j = 0; j < vec[i].size(); ++j) {
            out << vec[i][j];
            if (j + 1 != vec[j].size()) {
                out << " ";
            }
        }
```

```
        if (i != vec.size() - 1)
            out << e;
    }
    return out;
}



template<typename T>
void _dbg(const char *_s, T _h) { cerr << _s << " = " << _h << "\n"; }

template<typename T, typename... Ts>
void _dbg(const char *_s, T _h, Ts... _t) {
    for (int _b = 0; ((_b += *_s == '(') -= *_s == ')') != 0 || *_s != ',';)cerr << *_s++;
    cerr << " = " << _h << ",";
    _dbg(_s + 1, _t...);
}

#ifdef LOCAL
#define dbg(...) _dbg(#__VA_ARGS__, __VA_ARGS__)
#else
#define dbg(...)
#endif



int binpow(int a, int n) {
```

```
    int result = 1;
    while (n) {
        if (n & 1)
            result *= a;
        a *= a;
        n >>= 1;
    }
    return result;
}

/*
const int ML = 240;
char memory[ML << 20];
size_t ptr_ = 0;

void* operator new(size_t how_much) {
    ptr_ += how_much;
    assert(ptr_ < ML * 1024 * 1024);
    return memory + ptr_ - how_much;
}

void operator delete(void*) { }
*/
```

```
#define int int
int const INF = 2 * 1e9;
ll const LINF = 1e18;
const int MAXN = 4 * 100 * 1001; // 1e5 + 0
//ll const CONST = 9223372036854775807;
long double const PI = 3.14159265358979323846;
//int const MAXN = 101;
const int SQRT = 350;
const int LOG = 21;
const int MOD = 1e9 + 7;

vvi g;
int ssz = 0; int n;
set<vector<pii>> omegalul;
int cnt = 0;
void search_left(int x, int y, vector<pii> path, vvi used, int counter) {
    used[x][y] = 1;
    used[y][x] = 1;
    dbg(x, y);
    counter++;
    if (counter == ssz) {
        cnt++;
        vector<pii> path_2 = path, path_3 = path;
        if (y + 1 < n && !used[x][y + 1]) {
```

```
        path_2.eb(x, y + 1);
        sort(all(path_2));
        omegalul.emplace(path_2);
    }
    if (y - 1 > -1 && !used[x][y - 1]) {
        path_3.eb(x, y - 1);
        sort(all(path_3));
        omegalul.emplace(path_3);
    }
    sort(all(path));
    omegalul.emplace(path);
    return;
}
int x1 = y - 1;
int x2 = y + 1;
if (x1 > -1) {
    if (!used[x1][y]) {
        used[x1][y] = 1;
        used[y][x1] = 1;
        vector<pii> new_path1 = path;
        new_path1.eb(x1, y);
        if (x1 + 1 < n && g[x1][x1 + 1] && !used[x1][x1 + 1]) {
            used[x1][x1 + 1] = 1;
            used[x1 + 1][x1] = 1;
```

```
            search_left(x1, x1 + 1, new_path1, used, counter);
        }
        if (g[x1][x1] && !used[x1][x1]) {
            used[x1][x1] = 1;
            search_left(x1, x1, new_path1, used, counter);
        }
        if (x1 - 1 > -1 && g[x1][x1 - 1] && !used[x1][x1 - 1]) {
            used[x1][x1 - 1] = 1;
            used[x1 - 1][x1] = 1;
            search_left(x1, x1 - 1, new_path1, used, counter);
        }
        /*used.clear();
        path.clear();*/
    }
}
if (x2 < n) {
    x1 = x2;
    if (!used[x1][y]) {
        used[x1][y] = 1;
        used[y][x1] = 1;
        vector<pii> new_path1 = path;
        new_path1.eb(x1, y);
        if (x1 + 1 < n && g[x1][x1 + 1] && !used[x1][x1 + 1]) {
            used[x1][x1 + 1] = 1;
            used[x1 + 1][x1] = 1;
```

```
            search_left(x1, x1 + 1, new_path1, used, counter);
        }
        if (g[x1][x1] && !used[x1][x1]) {
            used[x1][x1] = 1;
            search_left(x1, x1, new_path1, used, counter);
        }
        if (x1 - 1 > -1 && g[x1][x1 - 1] && !used[x1][x1 - 1]) {
            used[x1][x1 - 1] = 1;
            used[x1 - 1][x1] = 1;
            search_left(x1, x1 - 1, new_path1, used, counter);
        }
        /*used.clear();
        path.clear();*/
      }
   }
}

void solve() {
   cin >> n;
   g.resize(n, vi(n));
   string t; cin >> t;
   string t2;
   vector<pii> edges;
   vector<pii> edges_rev;
   while (true) {
```

```
    cin >> t;
    if (t == "END") {
        break;
    }
    cin >> t2;
    int x = stoi(t);
    int y = stoi(t2);
    dbg(x, y);
    g[x - 1][y - 1] = 1;
    g[y - 1][x - 1] = 1;
    edges.eb(x - 1, y - 1);
    edges_rev.eb(y - 1, x - 1);
}
ssz = sz(edges);
vvi used(n, vi(n, 0));

for (auto & [x, y] : edges) {
    vector<pii> path;
    search_left(x, y, path, used, 0);
    if (x + 1 < n) {
        vector<pii> new_path1 = {{x, x + 1}};
        search_left(x, y, new_path1, used, 0);
    }
    if (x - 1 > -1) {
        vector<pii> new_path2 = {{x, x - 1}};
```

```
            search_left(x, y, new_path2, used, 0);
        }
    }
    dbg(sz(omegalul), cnt);

    cout << sz(omegalul);
}



signed main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    cout.tie(nullptr);
    solve();
    dbg(Time);
    return 0;
}
```

Задание 3. Попытка 3.

```
/*\YOURiSST_xD\*/
#include <iostream>
#include <string>
#include <vector>
#include <set>
#include <queue>
#include <deque>
#include <cmath>
#include <algorithm>
#include <numeric>
#include <map>
#include <chrono>
#include <random>
#include <bitset>
#include <unordered_set>
#include <unordered_map>
#include <cassert>

#define sz(a) ((int)((a).size()))
#define get(_type, _x) \
    _type _x; \
    cin >> _x
#define mp(a, b) make_pair(a, b)
#define e "\n"
```

```
#define pb push_back
#define eb emplace_back
#define ft first
#define sc second
#define vi vector<int>
#define vl vector<long long>
#define vvi vector<vector<int>>
#define vvl vector<vector<long long>>
#define pii pair<int, int>
#define pll pair<long long, long long>
#define all(a) (a).begin(), (a).end()
#define rall(a) (a).rbegin(), (a).rend()
#define max max<int>
#define min min<int>
#define Time (double)clock() / CLOCKS_PER_SEC
#define filein(FILE) freopen(FILE, "r", stdin)
#define fileout(FILE) freopen(FILE, "w", stdout)
typedef long long ll;
typedef long double ld;
typedef unsigned long long ull;
using namespace std;
mt19937 rng(chrono::steady_clock::now().time_since_epoch().count());
mt19937_64 rng_ll(chrono::steady_clock::now().time_since_epoch().count());

template<typename T1, typename T2>
```

```
ostream &operator<<(ostream &out, pair<T1, T2> p) {
    out << '<' << p.first << " | " << p.second << '>';
    return out;
}

template<typename T>
ostream &operator<<(ostream &out, const vector<T> &vec) {
    for (int i = 0; i < vec.size(); ++i) {
        out << vec[i];
        if (i + 1 != vec.size()) {
            out << " ";
        }
    }
    return out;
}

template<typename T>
ostream &operator<<(ostream &out, const vector<vector<T>> &vec) {
    for (int i = 0; i < vec.size(); ++i) {
        for (int j = 0; j < vec[i].size(); ++j) {
            out << vec[i][j];
            if (j + 1 != vec[j].size()) {
                out << " ";
            }
        }
```

```
        if (i != vec.size() - 1)
            out << e;
    }
    return out;
}



template<typename T>
void _dbg(const char *_s, T _h) { cerr << _s << " = " << _h << "\n"; }

template<typename T, typename... Ts>
void _dbg(const char *_s, T _h, Ts... _t) {
    for (int _b = 0; ((_b += *_s == '(') -= *_s == ')') != 0 || *_s != ',';)cerr << *_s++;
    cerr << " = " << _h << ",";
    _dbg(_s + 1, _t...);
}

#ifdef LOCAL
#define dbg(...) _dbg(#__VA_ARGS__, __VA_ARGS__)
#else
#define dbg(...)
#endif



int binpow(int a, int n) {
```

```
    int result = 1;
    while (n) {
        if (n & 1)
            result *= a;
        a *= a;
        n >>= 1;
    }
    return result;
}

/*
const int ML = 240;
char memory[ML << 20];
size_t ptr_ = 0;

void* operator new(size_t how_much) {
    ptr_ += how_much;
    assert(ptr_ < ML * 1024 * 1024);
    return memory + ptr_ - how_much;
}

void operator delete(void*) { }
*/
```

```
#define int int

int const INF = 2 * 1e9;

ll const LINF = 1e18;

const int MAXN = 4 * 100 * 1001; // 1e5 + 0

//ll const CONST = 9223372036854775807;

long double const PI = 3.14159265358979323846;

//int const MAXN = 101;

const int SQRT = 350;

const int LOG = 21;

const int MOD = 1e9 + 7;


vvi g;

int ssz = 0; int n;

set<vector<pii>> omegalul;

int cnt = 0;

void search_left(int x, int y, vector<pii> path, vvi used, int counter, vector<pii>&
old_path) {

    used[x][y] = 1;

    used[y][x] = 1;

    dbg(x, y);

    counter++;

    old_path.clear();

    if (counter == ssz) {

        cnt++;
```

```
    vector<pii> path_2 = path, path_3 = path;
    if (y + 1 < n && !used[x][y + 1]) {
        path_2.eb(x, y + 1);
        sort(all(path_2));
        omegalul.emplace(path_2);
    }
    if (y - 1 > -1 && !used[x][y - 1]) {
        path_3.eb(x, y - 1);
        sort(all(path_3));
        omegalul.emplace(path_3);
    }
    sort(all(path));
    omegalul.emplace(path);
    return;
}
int x1 = y - 1;
int x2 = y + 1;
if (x1 > -1) {
    if (!used[x1][y]) {
        used[x1][y] = 1;
        used[y][x1] = 1;
        vector<pii> new_path1 = path;
        new_path1.eb(x1, y);
        if (x1 + 1 < n && g[x1][x1 + 1] && !used[x1][x1 + 1]) {
            used[x1][x1 + 1] = 1;
```

```
            used[x1 + 1][x1] = 1;


            search_left(x1, x1 + 1, new_path1, used, counter, path);
        }
        if (g[x1][x1] && !used[x1][x1]) {
            used[x1][x1] = 1;
            search_left(x1, x1, new_path1, used, counter, path);
        }
        if (x1 - 1 > -1 && g[x1][x1 - 1] && !used[x1][x1 - 1]) {
            used[x1][x1 - 1] = 1;
            used[x1 - 1][x1] = 1;
            search_left(x1, x1 - 1, new_path1, used, counter, path);
        }
        /*used.clear();
        path.clear();*/
    }
}
if (x2 < n) {
    x1 = x2;
    if (!used[x1][y]) {
        used[x1][y] = 1;
        used[y][x1] = 1;
        vector<pii> new_path1 = path;
        new_path1.eb(x1, y);
        if (x1 + 1 < n && g[x1][x1 + 1] && !used[x1][x1 + 1]) {
```

```
            used[x1][x1 + 1] = 1;
            used[x1 + 1][x1] = 1;
            search_left(x1, x1 + 1, new_path1, used, counter, path);
        }
        if (g[x1][x1] && !used[x1][x1]) {
            used[x1][x1] = 1;
            search_left(x1, x1, new_path1, used, counter, path);
        }
        if (x1 - 1 > -1 && g[x1][x1 - 1] && !used[x1][x1 - 1]) {
            used[x1][x1 - 1] = 1;
            used[x1 - 1][x1] = 1;
            search_left(x1, x1 - 1, new_path1, used, counter, path);
        }

    }
  }
}

void solve() {
  cin >> n;
  g.resize(n, vi(n));
  string t; cin >> t;
  string t2;
  vector<pii> edges;
  vector<pii> edges_rev;
```

```
while (true) {
    cin >> t;
    if (t == "END") {
        break;
    }
    cin >> t2;
    int x = stoi(t);
    int y = stoi(t2);
    dbg(x, y);
    g[x - 1][y - 1] = 1;
    g[y - 1][x - 1] = 1;
    edges.eb(x - 1, y - 1);
    edges_rev.eb(y - 1, x - 1);
}
ssz = sz(edges);
vvi used(n, vi(n, 0));
vector<pii> temp = {{1, 42}, {4, 23}};
for (auto & [x, y] : edges) {
    vector<pii> path;
    search_left(x, y, path, used, 0, temp);
    if (x + 1 < n) {
        vector<pii> new_path1 = {{x, x + 1}};
        search_left(x, y, new_path1, used, 0, temp);
    }
    if (x - 1 > -1) {
```

```
            vector<pii> new_path2 = {{x, x - 1}};
            search_left(x, y, new_path2, used, 0, temp);
        }
    }
    dbg(sz(omegalul), cnt);

    cout << sz(omegalul);

}


signed main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    cout.tie(nullptr);
    solve();
    dbg(Time);
    return 0;
}


/*
10
FACE
1 1
```

2 2

3 3

4 4

5 5

6 5

6 7

7 6

7 8

7 7

8 8

9 9

10 10

11 11

END

*/

/*

100

ghjfn

1 1

2 2

3 3

4 4

5 5

6 6

7 7

8 8

9 9

10 10

11 11

12 12

13 13

14 14

15 15

16 16

17 17

18 18

19 19

20 20

21 21

22 22

23 23

24 24

25 25

26 26

27 27

28 28

29 29

30 30

31 31

32 32

33 33

34 34

35 35

36 36

37 37

38 38

39 39

40 40

41 41

42 42

43 43

44 44

45 45

46 46

47 47

48 48

49 49

50 50

51 51

52 52

53 53

54 54

55 55

56 56

57 57

58 58

59 59

60 60

61 61

62 62

63 63

64 64

65 65

66 66

67 67

68 68

69 69

70 70

71 71

72 72

73 73

74 74

75 75

76 76

77 77

78 78

79 79

80 80

81 81

82 82

83 83

84 84

85 85

86 86

87 87

88 88

89 89

90 90

91 91

92 92

93 93

94 94

95 95

96 96

97 97

98 98

99 99

END

*/

Задание 4. Попытка 1.

```
/*\YOURiSST_xD\*/
#include <iostream>
#include <string>
#include <vector>
#include <set>
#include <queue>
#include <deque>
#include <cmath>
#include <algorithm>
#include <numeric>
#include <map>
#include <chrono>
#include <random>
#include <bitset>
#include <unordered_set>
#include <unordered_map>
#include <cassert>

#define sz(a) ((int)((a).size()))
#define get(_type, _x) \
    _type _x; \
    cin >> _x
#define mp(a, b) make_pair(a, b)
#define e "\n"
```

```
#define pb push_back
#define eb emplace_back
#define ft first
#define sc second
#define vi vector<int>
#define vl vector<long long>
#define vvi vector<vector<int>>
#define vvl vector<vector<long long>>
#define pii pair<int, int>
#define pll pair<long long, long long>
#define all(a) (a).begin(), (a).end()
#define rall(a) (a).rbegin(), (a).rend()
#define max max<int>
#define min min<int>
#define Time (double)clock() / CLOCKS_PER_SEC
#define filein(FILE) freopen(FILE, "r", stdin)
#define fileout(FILE) freopen(FILE, "w", stdout)
typedef long long ll;
typedef long double ld;
typedef unsigned long long ull;
using namespace std;
mt19937 rng(chrono::steady_clock::now().time_since_epoch().count());
mt19937_64 rng_ll(chrono::steady_clock::now().time_since_epoch().count());

template<typename T1, typename T2>
```

```
ostream &operator<<(ostream &out, pair<T1, T2> p) {
    out << '<' << p.first << " | " << p.second << '>';
    return out;
}

template<typename T>
ostream &operator<<(ostream &out, const vector<T> &vec) {
    for (int i = 0; i < vec.size(); ++i) {
        out << vec[i];
        if (i + 1 != vec.size()) {
            out << " ";
        }
    }
    return out;
}

template<typename T>
ostream &operator<<(ostream &out, const vector<vector<T>> &vec) {
    for (int i = 0; i < vec.size(); ++i) {
        for (int j = 0; j < vec[i].size(); ++j) {
            out << vec[i][j];
            if (j + 1 != vec[j].size()) {
                out << " ";
            }
        }
```

```
        if (i != vec.size() - 1)
            out << e;
    }
    return out;
}



template<typename T>
void _dbg(const char *_s, T _h) { cerr << _s << " = " << _h << "\n"; }

template<typename T, typename... Ts>
void _dbg(const char *_s, T _h, Ts... _t) {
    for (int _b = 0; ((_b += *_s == '(') -= *_s == ')') != 0 || *_s != ',';)cerr << *_s++;
    cerr << " = " << _h << ",";
    _dbg(_s + 1, _t...);
}

#ifdef LOCAL
#define dbg(...) _dbg(#__VA_ARGS__, __VA_ARGS__)
#else
#define dbg(...)
#endif



int binpow(int a, int n) {
```

```
    int result = 1;
    while (n) {
        if (n & 1)
            result *= a;
        a *= a;
        n >>= 1;
    }
    return result;
}

/*
const int ML = 240;
char memory[ML << 20];
size_t ptr_ = 0;

void* operator new(size_t how_much) {
    ptr_ += how_much;
    assert(ptr_ < ML * 1024 * 1024);
    return memory + ptr_ - how_much;
}

void operator delete(void*) { }
*/
```

```
//#define int long long
int const INF = 2 * 1e9;
ll const LINF = 1e18;
const int MAXN = 4 * 100 * 1001; // 1e5 + 0
//ll const CONST = 9223372036854775807;
long double const PI = 3.14159265358979323846;
//int const MAXN = 101;
const int SQRT = 350;
const int LOG = 21;
const int MOD = 1e9 + 7;

struct Edge {
    int from, to;
    Edge(int from_, int to_) {
        from = from_;
        to = to_;
    }
};

vector<int> g[MAXN];
int h[MAXN];
int hup[MAXN];
bool used[MAXN];
vector<Edge> bridges;
```

```
void dfs(int v, int p) {
    h[v] = h[p] + 1;
    hup[v] = h[v];
    used[v] = true;
    for (auto & to : g[v]) {
        if (to == p) {
            continue;
        } else if (!used[to]) {
            dfs(to, v);
            hup[v] = min(hup[v], hup[to]);
            if (hup[to] > h[v]) {
                bridges.eb(v, to);
            }
        } else {
            hup[v] = min(hup[v], h[to]);
        }
    }
}



void solve() {
    int n, m; cin >> n >> m;

    while (m --> 0) {
```

```
    int u, v; cin >> u >> v;
    u--; v--;
    if (u == v)
        continue;
    g[u].eb(v);
    g[v].eb(u);
}
for (int i = 0; i < n; ++i) {
    if (!used[i])
        dfs(i, i);
}
dbg(sz(bridges));
if (sz(bridges) > 0) {
    cout << 1 << e;
    auto & [u, v] = *bridges.begin();
    if (u > v)
        swap(u, v);
    cout << u + 1 << " " << v + 1;
    exit(0);
}
int min_ans = INF;
vi vtxs;
int v_ans = -1;
for (int v = 0; v < MAXN; ++v) {
    if (sz(g[v]) > 0 && sz(g[v]) < min_ans) {
```

```
            min_ans = sz(g[v]);
            v_ans = v;
        }
    }
    vtxs = g[v_ans];
    vector<pii> ans;
    for (auto & to : g[v_ans]) {
        if (v_ans > to) {
            ans.eb(to, v_ans);
        } else {
            ans.eb(v_ans, to);
        }
    }
    sort(all(ans));
    cout << min_ans << e;
    for (auto & [u, v] : ans) {
        cout << u + 1 << " " << v + 1 << e;
    }
}



signed main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
```

```
    cout.tie(nullptr);
    solve();
    dbg(Time);
    return 0;
}
```