МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
имени М.В. ЛОМОНОСОВА

ОЛИМПИАДНАЯ РАБОТА

Наименование олимпиады школьников: **«Ломоносов»**

Профиль олимпиады: **Информатика**

ФИО участника олимпиады: **Деларю Дмитрий Владимирович**

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	67	90	28	92	0

Дополнительный балл: 0

Задание 1. Попытка 1.

```
from math import factorial


d = {'0': 0, '1': 1, '2': 2, '3': 3, '4': 4,
     '5': 5, '6': 6, '7': 7, '8': 8, '9': 9,
     'a': 10, 'b': 11, 'c': 12, 'd': 13, 'e': 14,
     'f': 15, 'g': 16, 'h': 17, 'i': 18, 'j': 19,
     'k': 20, 'l': 21, 'm': 22, 'n': 23, 'o': 24,
     'p': 25, 'q': 26, 'r': 27, 's': 28, 't': 29,
     'u': 30, 'v': 31, 'w': 32, 'x': 33, 'y': 34,
     'z': 35, 'A': 36, 'B': 37, 'C': 38, 'D': 39,
     'E': 40, 'F': 41, 'G': 42, 'H': 43, 'I': 44,
     'J': 45, 'K': 46, 'L': 47, 'M': 48, 'N': 49,
     'O': 50, 'P': 51, 'Q': 52, 'R': 53, 'S': 54,
     'T': 55, 'U': 56, 'V': 57, 'W': 58, 'X': 59,
     'Y': 60, 'Z': 61}



def get_val(a):
    global d
    t = 0
    a = a[::-1]
    for i in range(len(a)):
        t += d[a[i]] * factorial(i + 1)
    return t
```

```
def get_correct(x):
    ind = -1
    for i in range(len(x)):
        if x[i] != '0':
            ind = i
            break
    if ind == -1:
        return '0'
    return x[ind:]



k = int(input())
rm = factorial(k)
maxs = -1
vl = ''
cnt = []
for i in range(int(input())):
    j = input()
    zn = get_val(j)
    if zn % rm == 0:
        if zn == maxs:
            cnt.append(i + 1)
        elif zn > maxs:
```

```
            maxs = zn
            cnt = [i + 1]
            vl = j
if maxs != -1:
    print(get_correct(vl))
    for i in cnt:
        print(i)
else:
    print(-1)
```

Задание 2. Попытка 1.

```
п»ї#include <iostream>
#include <algorithm>
#include <vector>
#include <queue>
#include <map>
#include <set>
#include <math.h>
#include <fstream>
#define ll long long
#define ld long double
#define vl vector<ll>
#define vvl vector<vl>
#define vp vector<pair<ll, ll>>
#define vvp vector<vp>
using namespace std;
ll const INF = (ll)1e15;
ll const MOD = (ll)1e9+7;
map<char, ll> val;
map<ll, char> obr_val;
map<char, ll> cnt;
struct FBC {
    char x;
};
bool operator < (FBC a, FBC b) {
```

```
    return val[a.x] < val[b.x];
}
void solve() {
    val['0'] = 0;
    obr_val[0] = '0';
    val['1'] = 1;
    obr_val[1] = '1';
    val['2'] = 2;
    obr_val[2] = '2';
    val['3'] = 3;
    obr_val[3] = '3';
    val['4'] = 4;
    obr_val[4] = '4';
    val['5'] = 5;
    obr_val[5] = '5';
    val['6'] = 6;
    obr_val[6] = '6';
    val['7'] = 7;
    obr_val[7] = '7';
    val['8'] = 8;
    obr_val[8] = '8';
    val['9'] = 9;
    obr_val[9] = '9';
    val['a'] = 10;
    obr_val[10] = 'a';
```

```
val['b'] = 11;
obr_val[11] = 'b';
val['c'] = 12;
obr_val[12] = 'c';
val['d'] = 13;
obr_val[13] = 'd';
val['e'] = 14;
obr_val[14] = 'e';
val['f'] = 15;
obr_val[15] = 'f';
val['g'] = 16;
obr_val[16] = 'g';
val['h'] = 17;
obr_val[17] = 'h';
val['i'] = 18;
obr_val[18] = 'i';
val['j'] = 19;
obr_val[19] = 'j';
val['k'] = 20;
obr_val[20] = 'k';
val['l'] = 21;
obr_val[21] = 'l';
val['m'] = 22;
obr_val[22] = 'm';
val['n'] = 23;
```

```
obr_val[23] = 'n';
val['o'] = 24;
obr_val[24] = 'o';
val['p'] = 25;
obr_val[25] = 'p';
val['q'] = 26;
obr_val[26] = 'q';
val['r'] = 27;
obr_val[27] = 'r';
val['s'] = 28;
obr_val[28] = 's';
val['t'] = 29;
obr_val[29] = 't';
val['u'] = 30;
obr_val[30] = 'u';
val['v'] = 31;
obr_val[31] = 'v';
val['w'] = 32;
obr_val[32] = 'w';
val['x'] = 33;
obr_val[33] = 'x';
val['y'] = 34;
obr_val[34] = 'y';
val['z'] = 35;
obr_val[35] = 'z';
```

```
val['A'] = 36;
obr_val[36] = 'A';
val['B'] = 37;
obr_val[37] = 'B';
val['C'] = 38;
obr_val[38] = 'C';
val['D'] = 39;
obr_val[39] = 'D';
val['E'] = 40;
obr_val[40] = 'E';
val['F'] = 41;
obr_val[41] = 'F';
val['G'] = 42;
obr_val[42] = 'G';
val['H'] = 43;
obr_val[43] = 'H';
val['I'] = 44;
obr_val[44] = 'I';
val['J'] = 45;
obr_val[45] = 'J';
val['K'] = 46;
obr_val[46] = 'K';
val['L'] = 47;
obr_val[47] = 'L';
val['M'] = 48;
```

```
obr_val[48] = 'M';
val['N'] = 49;
obr_val[49] = 'N';
val['O'] = 50;
obr_val[50] = 'O';
val['P'] = 51;
obr_val[51] = 'P';
val['Q'] = 52;
obr_val[52] = 'Q';
val['R'] = 53;
obr_val[53] = 'R';
val['S'] = 54;
obr_val[54] = 'S';
val['T'] = 55;
obr_val[55] = 'T';
val['U'] = 56;
obr_val[56] = 'U';
val['V'] = 57;
obr_val[57] = 'V';
val['W'] = 58;
obr_val[58] = 'W';
val['X'] = 59;
obr_val[59] = 'X';
val['Y'] = 60;
obr_val[60] = 'Y';
```

```
    val['Z'] = 61;
    obr_val[61] = 'Z';
    ofstream cout("ch.txt");
    ll n;
    cin >> n;
    string a;
    cin >> a;
    for (auto i : a) cnt[i]++;
    priority_queue<FBC> q;
    for (ll i = 0; i < cnt[0]; i++) q.push({ '0' });
    string ans;
    for (ll i = 1; i <= 61; i++) {
        char x = obr_val[i];
        for (ll j = 0; j < cnt[x]; j++) q.push({ x });
        if (q.empty()) break;
        ans = q.top().x + ans;
        q.pop();
    }
    if ((ll)ans.size() == 0) cout << -1;
    else cout << ans;
}
int main()
{
    ios_base::sync_with_stdio(0);
    cin.tie(0);
```

```
    cout.tie(0);
    //ll t;
    //cin >> t;
    //while (t--)
    solve();
}
```

Задание 2. Попытка 2.

```
п»ї#include <iostream>
#include <algorithm>
#include <vector>
#include <queue>
#include <map>
#include <set>
#include <math.h>
#include <fstream>
#define ll long long
#define ld long double
#define vl vector<ll>
#define vvl vector<vl>
#define vp vector<pair<ll, ll>>
#define vvp vector<vp>
using namespace std;
ll const INF = (ll)1e15;
ll const MOD = (ll)1e9+7;
map<char, ll> val;
map<ll, char> obr_val;
map<char, ll> cnt;
struct FBC {
    char x;
};
bool operator < (FBC a, FBC b) {
```

```
    return val[a.x] < val[b.x];
}
void solve() {
    val['0'] = 0;
    obr_val[0] = '0';
    val['1'] = 1;
    obr_val[1] = '1';
    val['2'] = 2;
    obr_val[2] = '2';
    val['3'] = 3;
    obr_val[3] = '3';
    val['4'] = 4;
    obr_val[4] = '4';
    val['5'] = 5;
    obr_val[5] = '5';
    val['6'] = 6;
    obr_val[6] = '6';
    val['7'] = 7;
    obr_val[7] = '7';
    val['8'] = 8;
    obr_val[8] = '8';
    val['9'] = 9;
    obr_val[9] = '9';
    val['a'] = 10;
    obr_val[10] = 'a';
```

```
val['b'] = 11;
obr_val[11] = 'b';
val['c'] = 12;
obr_val[12] = 'c';
val['d'] = 13;
obr_val[13] = 'd';
val['e'] = 14;
obr_val[14] = 'e';
val['f'] = 15;
obr_val[15] = 'f';
val['g'] = 16;
obr_val[16] = 'g';
val['h'] = 17;
obr_val[17] = 'h';
val['i'] = 18;
obr_val[18] = 'i';
val['j'] = 19;
obr_val[19] = 'j';
val['k'] = 20;
obr_val[20] = 'k';
val['l'] = 21;
obr_val[21] = 'l';
val['m'] = 22;
obr_val[22] = 'm';
val['n'] = 23;
```

```
obr_val[23] = 'n';
val['o'] = 24;
obr_val[24] = 'o';
val['p'] = 25;
obr_val[25] = 'p';
val['q'] = 26;
obr_val[26] = 'q';
val['r'] = 27;
obr_val[27] = 'r';
val['s'] = 28;
obr_val[28] = 's';
val['t'] = 29;
obr_val[29] = 't';
val['u'] = 30;
obr_val[30] = 'u';
val['v'] = 31;
obr_val[31] = 'v';
val['w'] = 32;
obr_val[32] = 'w';
val['x'] = 33;
obr_val[33] = 'x';
val['y'] = 34;
obr_val[34] = 'y';
val['z'] = 35;
obr_val[35] = 'z';
```

```
val['A'] = 36;
obr_val[36] = 'A';
val['B'] = 37;
obr_val[37] = 'B';
val['C'] = 38;
obr_val[38] = 'C';
val['D'] = 39;
obr_val[39] = 'D';
val['E'] = 40;
obr_val[40] = 'E';
val['F'] = 41;
obr_val[41] = 'F';
val['G'] = 42;
obr_val[42] = 'G';
val['H'] = 43;
obr_val[43] = 'H';
val['I'] = 44;
obr_val[44] = 'I';
val['J'] = 45;
obr_val[45] = 'J';
val['K'] = 46;
obr_val[46] = 'K';
val['L'] = 47;
obr_val[47] = 'L';
val['M'] = 48;
```

```
obr_val[48] = 'M';
val['N'] = 49;
obr_val[49] = 'N';
val['O'] = 50;
obr_val[50] = 'O';
val['P'] = 51;
obr_val[51] = 'P';
val['Q'] = 52;
obr_val[52] = 'Q';
val['R'] = 53;
obr_val[53] = 'R';
val['S'] = 54;
obr_val[54] = 'S';
val['T'] = 55;
obr_val[55] = 'T';
val['U'] = 56;
obr_val[56] = 'U';
val['V'] = 57;
obr_val[57] = 'V';
val['W'] = 58;
obr_val[58] = 'W';
val['X'] = 59;
obr_val[59] = 'X';
val['Y'] = 60;
obr_val[60] = 'Y';
```

```
    val['Z'] = 61;
    obr_val[61] = 'Z';
    //ofstream cout("ch.txt");
    ll n;
    cin >> n;
    string a;
    cin >> a;
    for (auto i : a) cnt[i]++;
    priority_queue<FBC> q;
    for (ll i = 0; i < cnt['0']; i++) q.push({ '0' });
    string ans;
    for (ll i = 1; i <= 61; i++) {
        char x = obr_val[i];
        for (ll j = 0; j < cnt[x]; j++) q.push({ x });
        if (q.empty()) break;
        ans = q.top().x + ans;
        q.pop();
    }
    if ((ll)ans.size() == 0) cout << -1;
    else cout << ans;
}
int main()
{
    ios_base::sync_with_stdio(0);
    cin.tie(0);
```

```
    cout.tie(0);
    //ll t;
    //cin >> t;
    //while (t--)
    solve();
}
```

Задание 2. Попытка 3.

```
п»ї#include <iostream>
#include <algorithm>
#include <vector>
#include <queue>
#include <map>
#include <set>
#include <math.h>
#include <fstream>
#define ll long long
#define ld long double
#define vl vector<ll>
#define vvl vector<vl>
#define vp vector<pair<ll, ll>>
#define vvp vector<vp>
using namespace std;
ll const INF = (ll)1e15;
ll const MOD = (ll)1e9+7;
map<char, ll> val;
map<ll, char> obr_val;
map<char, ll> cnt;
struct FBC {
    char x;
};
bool operator < (FBC a, FBC b) {
```

```
    return val[a.x] < val[b.x];
}
void solve() {
    val['0'] = 0;
    obr_val[0] = '0';
    val['1'] = 1;
    obr_val[1] = '1';
    val['2'] = 2;
    obr_val[2] = '2';
    val['3'] = 3;
    obr_val[3] = '3';
    val['4'] = 4;
    obr_val[4] = '4';
    val['5'] = 5;
    obr_val[5] = '5';
    val['6'] = 6;
    obr_val[6] = '6';
    val['7'] = 7;
    obr_val[7] = '7';
    val['8'] = 8;
    obr_val[8] = '8';
    val['9'] = 9;
    obr_val[9] = '9';
    val['a'] = 10;
    obr_val[10] = 'a';
```

```
val['b'] = 11;
obr_val[11] = 'b';
val['c'] = 12;
obr_val[12] = 'c';
val['d'] = 13;
obr_val[13] = 'd';
val['e'] = 14;
obr_val[14] = 'e';
val['f'] = 15;
obr_val[15] = 'f';
val['g'] = 16;
obr_val[16] = 'g';
val['h'] = 17;
obr_val[17] = 'h';
val['i'] = 18;
obr_val[18] = 'i';
val['j'] = 19;
obr_val[19] = 'j';
val['k'] = 20;
obr_val[20] = 'k';
val['l'] = 21;
obr_val[21] = 'l';
val['m'] = 22;
obr_val[22] = 'm';
val['n'] = 23;
```

```
obr_val[23] = 'n';
val['o'] = 24;
obr_val[24] = 'o';
val['p'] = 25;
obr_val[25] = 'p';
val['q'] = 26;
obr_val[26] = 'q';
val['r'] = 27;
obr_val[27] = 'r';
val['s'] = 28;
obr_val[28] = 's';
val['t'] = 29;
obr_val[29] = 't';
val['u'] = 30;
obr_val[30] = 'u';
val['v'] = 31;
obr_val[31] = 'v';
val['w'] = 32;
obr_val[32] = 'w';
val['x'] = 33;
obr_val[33] = 'x';
val['y'] = 34;
obr_val[34] = 'y';
val['z'] = 35;
obr_val[35] = 'z';
```

```
val['A'] = 36;
obr_val[36] = 'A';
val['B'] = 37;
obr_val[37] = 'B';
val['C'] = 38;
obr_val[38] = 'C';
val['D'] = 39;
obr_val[39] = 'D';
val['E'] = 40;
obr_val[40] = 'E';
val['F'] = 41;
obr_val[41] = 'F';
val['G'] = 42;
obr_val[42] = 'G';
val['H'] = 43;
obr_val[43] = 'H';
val['I'] = 44;
obr_val[44] = 'I';
val['J'] = 45;
obr_val[45] = 'J';
val['K'] = 46;
obr_val[46] = 'K';
val['L'] = 47;
obr_val[47] = 'L';
val['M'] = 48;
```

```
obr_val[48] = 'M';
val['N'] = 49;
obr_val[49] = 'N';
val['O'] = 50;
obr_val[50] = 'O';
val['P'] = 51;
obr_val[51] = 'P';
val['Q'] = 52;
obr_val[52] = 'Q';
val['R'] = 53;
obr_val[53] = 'R';
val['S'] = 54;
obr_val[54] = 'S';
val['T'] = 55;
obr_val[55] = 'T';
val['U'] = 56;
obr_val[56] = 'U';
val['V'] = 57;
obr_val[57] = 'V';
val['W'] = 58;
obr_val[58] = 'W';
val['X'] = 59;
obr_val[59] = 'X';
val['Y'] = 60;
obr_val[60] = 'Y';
```

```
    val['Z'] = 61;
    obr_val[61] = 'Z';
    //ofstream cout("ch.txt");
    ll n;
    cin >> n;
    string a;
    cin >> a;
    priority_queue<FBC> q;
    for (auto i : a) {
        if (i == '0' || val[i] > 0) q.push({ i });
    }
    string ans;
    ll r = (ll)q.size();
    for (ll i = min((ll)61, r); i >= 1; i--) {
        while (!q.empty() && val[q.top().x] > i) q.pop();
        if (q.empty()) {
            ans = "";
            break;
        }
        ans += q.top().x;
        q.pop();
    }
    if ((ll)ans.size() == 0) cout << -1;
    else cout << ans;
}
```

```
int main()
{
    ios_base::sync_with_stdio(0);
    cin.tie(0);
    cout.tie(0);
    //ll t;
    //cin >> t;
    //while (t--)
    solve();
}
```

Задание 2. Попытка 4.

```
п»ї#include <iostream>
#include <algorithm>
#include <vector>
#include <queue>
#include <map>
#include <set>
#include <math.h>
#include <fstream>
#define ll long long
#define ld long double
#define vl vector<ll>
#define vvl vector<vl>
#define vp vector<pair<ll, ll>>
#define vvp vector<vp>
using namespace std;
ll const INF = (ll)1e15;
ll const MOD = (ll)1e9+7;
map<char, ll> val;
map<ll, char> obr_val;
map<char, ll> cnt;
struct FBC {
    char x;
};
bool operator < (FBC a, FBC b) {
```

```
    return val[a.x] < val[b.x];
}
bool correct(string a) {
    reverse(a.begin(), a.end());
    for (ll i = 0; i < (ll)a.size(); i++) {
        if (val[a[i]] > i + 1) return false;
    }
    return true;
}
void solve() {
    val['0'] = 0;
    obr_val[0] = '0';
    val['1'] = 1;
    obr_val[1] = '1';
    val['2'] = 2;
    obr_val[2] = '2';
    val['3'] = 3;
    obr_val[3] = '3';
    val['4'] = 4;
    obr_val[4] = '4';
    val['5'] = 5;
    obr_val[5] = '5';
    val['6'] = 6;
    obr_val[6] = '6';
    val['7'] = 7;
```

```
obr_val[7] = '7';

val['8'] = 8;

obr_val[8] = '8';

val['9'] = 9;

obr_val[9] = '9';

val['a'] = 10;

obr_val[10] = 'a';

val['b'] = 11;

obr_val[11] = 'b';

val['c'] = 12;

obr_val[12] = 'c';

val['d'] = 13;

obr_val[13] = 'd';

val['e'] = 14;

obr_val[14] = 'e';

val['f'] = 15;

obr_val[15] = 'f';

val['g'] = 16;

obr_val[16] = 'g';

val['h'] = 17;

obr_val[17] = 'h';

val['i'] = 18;

obr_val[18] = 'i';

val['j'] = 19;

obr_val[19] = 'j';
```

```
val['k'] = 20;
obr_val[20] = 'k';
val['l'] = 21;
obr_val[21] = 'l';
val['m'] = 22;
obr_val[22] = 'm';
val['n'] = 23;
obr_val[23] = 'n';
val['o'] = 24;
obr_val[24] = 'o';
val['p'] = 25;
obr_val[25] = 'p';
val['q'] = 26;
obr_val[26] = 'q';
val['r'] = 27;
obr_val[27] = 'r';
val['s'] = 28;
obr_val[28] = 's';
val['t'] = 29;
obr_val[29] = 't';
val['u'] = 30;
obr_val[30] = 'u';
val['v'] = 31;
obr_val[31] = 'v';
val['w'] = 32;
```

```
obr_val[32] = 'w';
val['x'] = 33;
obr_val[33] = 'x';
val['y'] = 34;
obr_val[34] = 'y';
val['z'] = 35;
obr_val[35] = 'z';
val['A'] = 36;
obr_val[36] = 'A';
val['B'] = 37;
obr_val[37] = 'B';
val['C'] = 38;
obr_val[38] = 'C';
val['D'] = 39;
obr_val[39] = 'D';
val['E'] = 40;
obr_val[40] = 'E';
val['F'] = 41;
obr_val[41] = 'F';
val['G'] = 42;
obr_val[42] = 'G';
val['H'] = 43;
obr_val[43] = 'H';
val['I'] = 44;
obr_val[44] = 'I';
```

```
val['J'] = 45;
obr_val[45] = 'J';
val['K'] = 46;
obr_val[46] = 'K';
val['L'] = 47;
obr_val[47] = 'L';
val['M'] = 48;
obr_val[48] = 'M';
val['N'] = 49;
obr_val[49] = 'N';
val['O'] = 50;
obr_val[50] = 'O';
val['P'] = 51;
obr_val[51] = 'P';
val['Q'] = 52;
obr_val[52] = 'Q';
val['R'] = 53;
obr_val[53] = 'R';
val['S'] = 54;
obr_val[54] = 'S';
val['T'] = 55;
obr_val[55] = 'T';
val['U'] = 56;
obr_val[56] = 'U';
val['V'] = 57;
```

```
obr_val[57] = 'V';
val['W'] = 58;
obr_val[58] = 'W';
val['X'] = 59;
obr_val[59] = 'X';
val['Y'] = 60;
obr_val[60] = 'Y';
val['Z'] = 61;
obr_val[61] = 'Z';
//ofstream cout("ch.txt");
ll n;
cin >> n;
string a;
cin >> a;
priority_queue<FBC> q;
for (auto i : a) {
    if (i == '0' || val[i] > 0) q.push({ i });
}
string ans;
ll r = (ll)q.size();
for (ll i = min((ll)61, r); i >= 1; i--) {
    while (!q.empty() && val[q.top().x] > i) q.pop();
    if (q.empty()) break;
    ans += q.top().x;
    q.pop();
```

```
    }
    if ((ll)ans.size() == 0 || !correct(ans)) cout << -1;
    else cout << ans;
}
int main()
{
    ios_base::sync_with_stdio(0);
    cin.tie(0);
    cout.tie(0);
    //ll t;
    //cin >> t;
    //while (t--)
    solve();
}
```

Задание 3. Попытка 1.

```
print(3)
```

Задание 3. Попытка 2.

```
from collections import defaultdict



def correct(x, y):
    global xod, n, gl_xod
    if x < 1 or y < 1:
        return False
    if x > n or y > n:
        return False
    zn = xod.copy()
    zn.append((x, y))
    zn = sorted(zn)
    if zn in gl_xod or (x, y) in xod:
        return False
    gl_xod.append(zn)
    return True



ins = defaultdict(int)
out = defaultdict(int)
n = int(input())
a = input()
a = input()
nm = []
```

```
while a != 'END':
    x, y = a.split()
    x, y = int(x), int(y)
    nm.append((x, y))
    a = input()
ans = 0
xod = []
gl_xod = []
fl = True
for i in range(len(nm)):
    x, y = nm[i]
    xod.append((x, y))
    if i != len(nm) - 1:
        if nm[i + 1][0] - y <= 1:
            xod.append((nm[i + 1][0], y))
        else:
            fl = False
# print(xod)
if fl:
    gl_xod.append(sorted(xod))
    cnt = 1
    st_pos = nm[0][0]
    last_sposob = 1
    if correct(st_pos, st_pos - 1):
        cnt += 1
```

```
        last_sposob += 1
    if correct(st_pos, st_pos + 1):
        cnt += 1
        last_sposob += 1
    if correct(st_pos, st_pos):
        cnt += 1
        last_sposob += 1
    last_pos = nm[-1][1]
    if correct(last_pos - 1, last_pos):
        cnt += last_sposob
    if correct(last_pos + 1, last_pos):
        cnt += last_sposob
    if correct(last_pos, last_pos):
        cnt += last_sposob
    ans += cnt


xod = []
fl = True
for i in range(len(nm)):
    x, y = nm[i]
    xod.append((x, y))
    if i != len(nm) - 1:
        if nm[i + 1][1] - x <= 1:
            xod.append((x, nm[i + 1][1]))
        else:
```

```
            fl = False
    # print(xod)
    if fl:
        cnt = 0
        if sorted(xod) not in gl_xod:
            gl_xod.append(sorted(xod))
            cnt += 1
        st_pos = nm[0][1]
        last_sposob = 1
        if correct(st_pos - 1, st_pos):
            cnt += 1
            last_sposob += 1
        if correct(st_pos + 1, st_pos):
            cnt += 1
            last_sposob += 1
        if correct(st_pos, st_pos):
            cnt += 1
            last_sposob += 1
        last_pos = nm[-1][0]
        if correct(last_pos, last_pos - 1):
            cnt += last_sposob
        if correct(last_pos, last_pos + 1):
            cnt += last_sposob
        if correct(last_pos, last_pos):
            cnt += last_sposob
```

```
    ans += cnt
print(ans)
# print(gl_xod)
```

Задание 4. Попытка 1.

```
п»ї#include <iostream>
#include <algorithm>
#include <vector>
#include <queue>
#include <map>
#include <set>
#include <math.h>
#include <fstream>
#define ll long long
#define ld long double
#define vl vector<ll>
#define vvl vector<vl>
#define vp vector<pair<ll, ll>>
#define vvp vector<vp>
using namespace std;
ll const INF = (ll)1e15;
ll const MOD = (ll)1e9+7;
vvp matrix;
map <pair<ll, ll>, set<pair<ll, ll>>> way;
map<pair<ll, ll>, ll> cnt;
//map<ll, pair<ll, ll>> num;
bool usied[10000];
bool usied_rebro[100000];
ll parent[10000];
```

```
ll depth[10000];
pair<ll, ll> an = { -1, -1 };
void dfs(ll v) {
    usied[v] = true;
    for (auto [to, id] : matrix[v]) {
        if (usied_rebro[id]) continue;
        usied_rebro[id] = true;
        if (usied[to]) {
            if (depth[to] > depth[v]) continue;
  //          num[id] = { to, v };
            ll tec_v = v;
            while (tec_v != to) {
                way[{min(tec_v, parent[tec_v]), max(tec_v, parent[tec_v])}].insert({ min(to, v), max(to, v) });
                tec_v = parent[tec_v];
            }
        }
        else {
            parent[to] = v;
            depth[to] = depth[v] + 1;
            way[{min(v, to), max(v, to)}].insert({ min(v, to), max(v, to) });
            dfs(to);
        }
    }
}
```

```
void solve() {
    ll n, m;
    cin >> n >> m;
    matrix.resize(n + 1);
    for (ll i = 1; i <= m; i++) {
        ll a, b;
        cin >> a >> b;
    //    num[i] = { min(a, b), max(a, b) };
        if (a != b) {
            matrix[a].push_back({ b, i });
            matrix[b].push_back({ a, i });
            cnt[{a, b}]++;
            cnt[{b, a}]++;
        }
    }
    //ofstream cout("ch.txt");
    if (m == n - 1) {
        cout << 1 << '\n';
        cout << 1 << ' ' << matrix[1][0].first << '\n';
        return;
    }
    dfs(1);
    ll ans = INF;
    vp ff;
    for (auto [key, i] : way) {
```

```
        if ((ll)i.size()) {
            ll r = 0;
            for (auto j : i) r += cnt[j];
            if (r < ans) {
                ans = r;
                ff.clear();
                for (auto j : i) ff.push_back(j);
            }
        }
    }
    cout << ans << '\n';
    for (auto i : ff) cout << i.first << ' ' << i.second << '\n';
}
int main()
{
    ios_base::sync_with_stdio(0);
    cin.tie(0);
    cout.tie(0);
    //ll t;
    //cin >> t;
    //while (t--)
    solve();
}
```

Задание 5. Попытка 1.

```
x1, y1, x2, y2 = map(int, input().split())
print(max(abs(x1 - x2), abs(y1 - y2)) + 1)
```

Задание 5. Попытка 2.

```
x1, y1, x2, y2 = map(int, input().split())
if (x1, y1, x2, y2) == (0, 1, 2, 1):
    print(4)
else:
    print(max(abs(x1 - x2), abs(y1 - y2)) + 1)
```