



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

**ОЛИМПИАДНАЯ РАБОТА**

Наименование олимпиады школьников: «**Ломоносов**»

Профиль олимпиады: **Информатика**

ФИО участника олимпиады: **Юхневич Егор Владимирович**

Класс: **11 класс**

Технический балл: **99**

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	76	27	100	66

Дополнительный балл: 55

### Задание 1. Попытка 1.

```
#include <bits/stdc++.h>

using namespace std;

using u8 = uint8_t;
using u16 = uint16_t;
using u32 = uint32_t;
using u64 = uint64_t;

using s8 = int8_t;
using s16 = int16_t;
using s32 = int32_t;
using s64 = int64_t;

using ld = long double;

#define all(obj) obj.begin(), obj.end()

/*
 * SOLVE PROBLEM
 */

mt19937_64 rnd(42);

char getch(int i) {
```

```
if (i <= 9) {
    return i + '0';
}
else if (i <= 35) {
    return i - 10 + 'a';
}
else if(i <= 61) {
    return i - 10 - 26 + 'A';
}
else {
    cout << "getch\n";
    return '#';
}
}

int getval(char c) {
    if ('0' <= c && c <= '9') {
        return c - '0';
    }
    else if ('a' <= c && c <= 'z') {
        return c - 'a' + 10;
    }
    else if ('A' <= c && c <= 'Z') {
        return c - 'A' + 10 + 26;
    }
}
```

```

else {
    cout << "getval\n";
    return -1;
}
}

string get_rnd_word() {
    int len = rnd() % 10 + 1;
    string s;
    for(int z = 1; z <= len; z++) {
        s += getch(rnd() % (z + 1));
    }
    return s;
}

u64 cast(string s) {
    int n = s.size();
    u64 fact = 1;
    u64 d = 0;
    for (int i = 0; i < n; i++) {
        fact *= (i + 1);
        d += fact * getch(s[i]);
    }
    return d;
}

```

```
u64 fact(u64 n) {  
    u64 res = 1;  
    for (u64 x = 2; x <= n; x++) {  
        res *= x;  
    }  
    return res;  
}
```

```
bool trivial_check(string s, int k) {  
    u64 x = cast(s);  
    return x % fact(k) == 0;  
}
```

```
bool smart_check(string s, int k) {  
    for (int i = 0; i < min((int)s.size(), k - 1); i++) {  
        if (s[i] != '0') {  
            return false;  
        }  
    }  
    return true;  
}
```

```
int trivial_comp(string lhs, string rhs) {  
    return cast(lhs) < cast(rhs) ? -1 : (cast(lhs) == cast(rhs) ? 0 : +1);  
}
```

```
}
```

```
void rlz(string& s) {  
    while (!s.empty() && s.back() == '0') {  
        s.pop_back();  
    }  
    if (s.empty()) {  
        s += '0';  
    }  
}
```

```
// lhs == rhs: 0
```

```
// lhs < rhs: -1
```

```
// lhs > rhs: +1
```

```
int smart_comp(string lhs, string rhs) {  
    rlz(lhs);  
    rlz(rhs);  
  
    if (lhs.size() != rhs.size()) {  
        return lhs.size() < rhs.size() ? -1 : +1;  
    }  
    else {  
        for (int i = (int)lhs.size() - 1; i >= 0; i--) {  
            if (lhs[i] != rhs[i]) {  
                return lhs[i] < rhs[i] ? -1 : +1;  
            }  
        }  
    }  
}
```

```
    }  
  }  
  return 0; // lhs == rhs  
}  
}
```

```
void solve(istream& cin, ostream& cout) {  
  int k, n;  
  cin >> k >> n;  
  string max = "0";  
  vector<int> pos;  
  for (int i = 0; i < n; i++) {  
    string s;  
    cin >> s;  
    reverse(all(s));  
  
    if (smart_check(s, k)) {  
      if (smart_comp(max, s) == -1) {  
        max = s;  
        pos = { i };  
      }  
      else if (smart_comp(max, s) == 0){  
        pos.push_back(i);  
      }  
    }  
  }  
}
```



```
    }  
    rlz(max);  
    reverse(all(max));  
    cout << max << "\n";  
    for (int x : pos) {  
        cout << x + 1 << "\n";  
    }  
}  
  
/*  
* TEMPLATE MAIN  
*/  
  
int main() {  
    ifstream cin("input.txt");  
    ios::sync_with_stdio(false), cout.tie(nullptr), cin.tie(nullptr);  
  
    s64 t = 1;  
    //cin >> t;  
    while (t--) {  
        solve(cin, cout);  
    }  
    return 0;  
}
```



## Задание 1. Попытка 2.

```
#include <bits/stdc++.h>

using namespace std;

using u8 = uint8_t;
using u16 = uint16_t;
using u32 = uint32_t;
using u64 = uint64_t;

using s8 = int8_t;
using s16 = int16_t;
using s32 = int32_t;
using s64 = int64_t;

using ld = long double;

#define all(obj) obj.begin(), obj.end()

/*
 * SOLVE PROBLEM
 */

mt19937_64 rnd(42);

char getch(int i) {
```

```
if (i <= 9) {
    return i + '0';
}
else if (i <= 35) {
    return i - 10 + 'a';
}
else if(i <= 61) {
    return i - 10 - 26 + 'A';
}
else {
    cout << "getch\n";
    return '#';
}
}

int getval(char c) {
    if ('0' <= c && c <= '9') {
        return c - '0';
    }
    else if ('a' <= c && c <= 'z') {
        return c - 'a' + 10;
    }
    else if ('A' <= c && c <= 'Z') {
        return c - 'A' + 10 + 26;
    }
}
```

```

else {
    cout << "getval\n";
    return -1;
}
}

string get_rnd_word() {
    int len = rnd() % 10 + 1;
    string s;
    for(int z = 1; z <= len; z++) {
        s += getch(rnd() % (z + 1));
    }
    return s;
}

u64 cast(string s) {
    int n = s.size();
    u64 fact = 1;
    u64 d = 0;
    for (int i = 0; i < n; i++) {
        fact *= (i + 1);
        d += fact * getch(s[i]);
    }
    return d;
}

```

```
u64 fact(u64 n) {  
    u64 res = 1;  
    for (u64 x = 2; x <= n; x++) {  
        res *= x;  
    }  
    return res;  
}
```

```
bool trivial_check(string s, int k) {  
    u64 x = cast(s);  
    return x % fact(k) == 0;  
}
```

```
bool smart_check(string s, int k) {  
    for (int i = 0; i < min((int)s.size(), k - 1); i++) {  
        if (s[i] != '0') {  
            return false;  
        }  
    }  
    return true;  
}
```

```
int trivial_comp(string lhs, string rhs) {  
    return cast(lhs) < cast(rhs) ? -1 : (cast(lhs) == cast(rhs) ? 0 : +1);  
}
```

```
}
```

```
void rlz(string& s) {  
    while (!s.empty() && s.back() == '0') {  
        s.pop_back();  
    }  
    if (s.empty()) {  
        s += '0';  
    }  
}
```

```
// lhs == rhs: 0
```

```
// lhs < rhs: -1
```

```
// lhs > rhs: +1
```

```
int smart_comp(string lhs, string rhs) {  
    rlz(lhs);  
    rlz(rhs);  
  
    if (lhs.size() != rhs.size()) {  
        return lhs.size() < rhs.size() ? -1 : +1;  
    }  
    else {  
        for (int i = (int)lhs.size() - 1; i >= 0; i--) {  
            if (lhs[i] != rhs[i]) {  
                return getval(lhs[i]) < getval(rhs[i]) ? -1 : +1;  
            }  
        }  
    }  
}
```

```
    }  
  }  
  return 0; // lhs == rhs  
}  
}
```

```
void solve(istream& cin, ostream& cout) {  
  int k, n;  
  cin >> k >> n;  
  string max = "0";  
  vector<int> pos;  
  for (int i = 0; i < n; i++) {  
    string s;  
    cin >> s;  
    reverse(all(s));  
  
    if (smart_check(s, k)) {  
      if (smart_comp(max, s) == -1) {  
        max = s;  
        pos = { i };  
      }  
      else if (smart_comp(max, s) == 0){  
        pos.push_back(i);  
      }  
    }  
  }  
}
```



```
    }  
    rlz(max);  
    reverse(all(max));  
    cout << max << "\n";  
    for (int x : pos) {  
        cout << x + 1 << "\n";  
    }  
}  
  
/*  
* TEMPLATE MAIN  
*/  
  
int main() {  
    //ifstream cin("input.txt");  
    ios::sync_with_stdio(false), cout.tie(nullptr), cin.tie(nullptr);  
  
    s64 t = 1;  
    //cin >> t;  
    while (t--){  
        solve(cin, cout);  
    }  
    return 0;  
}
```



### Задание 1. Попытка 3.

```
#include <bits/stdc++.h>

using namespace std;

using u8 = uint8_t;
using u16 = uint16_t;
using u32 = uint32_t;
using u64 = uint64_t;

using s8 = int8_t;
using s16 = int16_t;
using s32 = int32_t;
using s64 = int64_t;

using ld = long double;

#define all(obj) obj.begin(), obj.end()

/*
 * SOLVE PROBLEM
 */

mt19937_64 rnd(42);

char getch(int i) {
```

```
if (i <= 9) {
    return i + '0';
}
else if (i <= 35) {
    return i - 10 + 'a';
}
else if(i <= 61) {
    return i - 10 - 26 + 'A';
}
else {
    cout << "getch\n";
    return '#';
}
}

int getval(char c) {
    if ('0' <= c && c <= '9') {
        return c - '0';
    }
    else if ('a' <= c && c <= 'z') {
        return c - 'a' + 10;
    }
    else if ('A' <= c && c <= 'Z') {
        return c - 'A' + 10 + 26;
    }
}
```

```
else {
    cout << "getval\n";
    return -1;
}
}

string get_rnd_word() {
    int len = rnd() % 10 + 1;
    string s;
    for(int z = 1; z <= len; z++) {
        s += getch(rnd() % (z + 1));
    }
    return s;
}

u64 cast(string s) {
    int n = s.size();
    u64 fact = 1;
    u64 d = 0;
    for (int i = 0; i < n; i++) {
        fact *= (i + 1);
        d += fact * getch(s[i]);
    }
    return d;
}
```

```
u64 fact(u64 n) {  
    u64 res = 1;  
    for (u64 x = 2; x <= n; x++) {  
        res *= x;  
    }  
    return res;  
}
```

```
bool trivial_check(string s, int k) {  
    u64 x = cast(s);  
    return x % fact(k) == 0;  
}
```

```
bool smart_check(string s, int k) {  
    for (int i = 0; i < min((int)s.size(), k - 1); i++) {  
        if (s[i] != '0') {  
            return false;  
        }  
    }  
    return true;  
}
```

```
int trivial_comp(string lhs, string rhs) {  
    return cast(lhs) < cast(rhs) ? -1 : (cast(lhs) == cast(rhs) ? 0 : +1);  
}
```

```
}
```

```
void rlz(string& s) {  
    while (!s.empty() && s.back() == '0') {  
        s.pop_back();  
    }  
    if (s.empty()) {  
        s += '0';  
    }  
}
```

```
// lhs == rhs: 0
```

```
// lhs < rhs: -1
```

```
// lhs > rhs: +1
```

```
int smart_comp(string lhs, string rhs) {  
    rlz(lhs);  
    rlz(rhs);  
  
    if (lhs.size() != rhs.size()) {  
        return lhs.size() < rhs.size() ? -1 : +1;  
    }  
    else {  
        for (int i = (int)lhs.size() - 1; i >= 0; i--) {  
            if (lhs[i] != rhs[i]) {  
                return getval(lhs[i]) < getval(rhs[i]) ? -1 : +1;  
            }  
        }  
    }  
}
```

```
    }  
  }  
  return 0; // lhs == rhs  
}  
}
```

```
void solve(istream& cin, ostream& cout) {  
  int k, n;  
  cin >> k >> n;  
  string max = "0";  
  vector<int> pos;  
  for (int i = 0; i < n; i++) {  
    string s;  
    cin >> s;  
    reverse(all(s));  
  
    if (smart_check(s, k)) {  
      if (smart_comp(max, s) == -1) {  
        max = s;  
        pos = { i };  
      }  
      else if (smart_comp(max, s) == 0){  
        pos.push_back(i);  
      }  
    }  
  }  
}
```



```
    }
    if (pos.empty()) {
        cout << "-1\n";
        return;
    }
    rlz(max);
    reverse(all(max));
    cout << max << "\n";
    for (int x : pos) {
        cout << x + 1 << "\n";
    }
}

/*
* TEMPLATE MAIN
*/

int main() {
    //ifstream cin("input.txt");
    ios::sync_with_stdio(false), cout.tie(nullptr), cin.tie(nullptr);

    s64 t = 1;
    //cin >> t;
    while (t--) {
```

```
    solve(cin, cout);  
}  
return 0;  
}
```

## Задание 2. Попытка 1.

```
#include <bits/stdc++.h>

using namespace std;

using u8 = uint8_t;
using u16 = uint16_t;
using u32 = uint32_t;
using u64 = uint64_t;

using s8 = int8_t;
using s16 = int16_t;
using s32 = int32_t;
using s64 = int64_t;

using ld = long double;

#define all(obj) obj.begin(), obj.end()

/*
 * SOLVE PROBLEM
 */

mt19937_64 rnd(42);

char getch(int i) {
```

```
if (i <= 9) {
    return i + '0';
}
else if (i <= 35) {
    return i - 10 + 'a';
}
else if(i <= 61) {
    return i - 10 - 26 + 'A';
}
else {
    return '#';
}
}

int getval(char c) {
    if ('0' <= c && c <= '9') {
        return c - '0';
    }
    else if ('a' <= c && c <= 'z') {
        return c - 'a' + 10;
    }
    else if ('A' <= c && c <= 'Z') {
        return c - 'A' + 10 + 26;
    }
    else {
```

```
    return -1;
}
}
```

```
string get_rnd_word() {
    int len = rnd() % 10 + 1;
    string s;
    for(int z = 1; z <= len; z++) {
        s += getch(rnd() % (z + 1));
    }
    return s;
}
```

```
u64 cast(string s) {
    int n = s.size();
    u64 fact = 1;
    u64 d = 0;
    for (int i = 0; i < n; i++) {
        fact *= (i + 1);
        d += fact * getch(s[i]);
    }
    return d;
}
```

```
u64 fact(u64 n) {
```

```
u64 res = 1;
for (u64 x = 2; x <= n; x++) {
    res *= x;
}
return res;
}
```

```
bool trivial_check(string s, int k) {
    u64 x = cast(s);
    return x % fact(k) == 0;
}
```

```
bool smart_check(string s, int k) {
    for (int i = 0; i < min((int)s.size(), k - 1); i++) {
        if (s[i] != '0') {
            return false;
        }
    }
    return true;
}
```

```
int trivial_comp(string lhs, string rhs) {
    return cast(lhs) < cast(rhs) ? -1 : (cast(lhs) == cast(rhs) ? 0 : +1);
}
```

```

void rlz(string& s) {
    while (!s.empty() && s.back() == '0') {
        s.pop_back();
    }
    if (s.empty()) {
        s += '0';
    }
}

// lhs == rhs: 0
// lhs < rhs: -1
// lhs > rhs: +1

int smart_comp(string lhs, string rhs) {
    rlz(lhs);
    rlz(rhs);

    if (lhs.size() != rhs.size()) {
        return lhs.size() < rhs.size() ? -1 : +1;
    }
    else {
        for (int i = (int)lhs.size() - 1; i >= 0; i--) {
            if (lhs[i] != rhs[i]) {
                return getval(lhs[i]) < getval(rhs[i]) ? -1 : +1;
            }
        }
    }
}

```

```
        return 0; // lhs == rhs
    }
}

void solve(istream& cin, ostream& cout) {
    vector<int> cnt(62);
    {
        int n;
        cin >> n;
        string s;
        cin >> s;
        n = s.size();

        for (char c : s) {
            if (getval(c) != -1) {
                cnt[getval(c)]++;
            }
        }
        for (int x = 0; x <= 61; x++) {
            if (cnt[x] > 61) {
                cnt[x] = 61;
            }
        }
    }
}
```



```

string s;

for (int x = 0; x <= 61; x++) {
    while (cnt[x]--) {
        s += getch(x);
    }
}

// C f P r P ° P » P ë C , C Ъ C , P μ , P e P s C , P s C Ъ C < P μ P S P μ P r P s P » P ¶ P S C < C f C , P s C ¶ C , C Ъ
P S P ° P r P ° P S P S P N ̄ P i P s P · P ë C † P ë P ë

auto del = [&]() {
    for (int i = 0; i < s.size(); i++) {
        if (i + 1 < getval(s[i])) {
            s.erase(s.begin() + i);
            i--;
        }
    }
};

del();

// C f P r P ° P » P ë C , C Ъ P » P ë C e P S P e P μ

while (s.size() > 61) {
    bool find = false;
    for (int i = 0; i + 1 < s.size(); i++) {
        if (getval(s[i + 1]) <= i + 1) {
            s.erase(s.begin() + i);
        }
    }
}

```

```
        find = true;
        break;
    }
}
if (!find) {
    s.pop_back();
}
}
if (s.empty()) {
    cout << "-1\n";
    return;
}
rlz(s);
reverse(all(s));
cout << s << "\n";
}

/*
* TEMPLATE MAIN
*/

int main() {
    //ifstream cin("input.txt");
    ios::sync_with_stdio(false), cout.tie(nullptr), cin.tie(nullptr);
```

```
s64 t = 1;
//cin >> t;
while (t-- > 0) {
    solve(cin, cout);
}
return 0;
}
```

### Задание 3. Попытка 1.

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
using u8 = uint8_t;
```

```
using u16 = uint16_t;
```

```
using u32 = uint32_t;
```

```
using u64 = uint64_t;
```

```
using s8 = int8_t;
```

```
using s16 = int16_t;
```

```
using s32 = int32_t;
```

```
using s64 = int64_t;
```

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

```
#define all(obj) obj.begin(), obj.end()
```

```
/*
```

```
* SOLVE PROBLEM
```

```
*/
```

```
int n;
```

```
string L;
```

```
vector<bool> visL, visR;
```

```
void paint(int v, bool is_left, string& R) {
```

```
    if(is_left) {
```

```
        if (visL[v]) {
```

```
            return;
```

```
        }
```

```
        visL[v] = true;
```

```
        if (L[v] == 'R') {
```

```
            paint(v, !is_left, R);
```

```
        }
```

```
        else if (L[v] == 'U') {
```

```
            paint(v - 1, !is_left, R);
```

```
        }
```

```
        else if (L[v] == 'D') {
```

```
            paint(v + 1, !is_left, R);
```

```
        }
```

```
    }
```

```
    else {
```

```
        if (visR[v]) {
```

```
            return;
```

```
        }
```

```
        visR[v] = true;
```

```

if (R[v] == 'L') {
    paint(v, !is_left, R);
}
else if (R[v] == 'U') {
    paint(v - 1, !is_left, R);
}
else if (R[v] == 'D') {
    paint(v + 1, !is_left, R);
}
}
}

```

```

bool verify(string R) {
    for (int i = 0; i < n; i++) {
        visL.assign(n, false);
        visR.assign(n, false);
        paint(i, true, R);

        bool ok = true;
        for (int i = 0; i < n && ok; i++) {
            if (L[i] != '#' && !visL[i]) {
                ok = false;
            }
            if (R[i] != '#' && !visR[i]) {

```

```
        ok = false;
    }
}

if (ok) {
    return true;
}
}
return false;
}

set<string> ss;

void f(int v, string& R) {
    if (verify(R)) {
        ss.insert(R);
    }

    if (R[v] != '#') {
        return;
    }

    if (L[v] == 'R' && R[v] == '#') {
        // U or D
    }
}
```

```

// U
if (v > 0 && L[v - 1] != '#') {
    R[v] = 'U';
    f(v - 1, R);
}

// D
if (v + 1 < n && L[v + 1] != '#') {
    R[v] = 'D';
    f(v + 1, R);
}

R[v] = '#';
}

else if (L[v] == 'U' && R[v - 1] == '#') {
    // L or U

    // L
    if (L[v - 1] != '#') {
        R[v - 1] = 'L';
        f(v - 1, R);
    }

    // U
    if (v - 2 >= 0 && L[v - 2] != '#') {
        R[v - 1] = 'U';

```



```

        f(v - 2, R);
    }
    R[v - 1] = '#';
}
else if (L[v] == 'D' && R[v + 1] == '#') {
    // L or D

    // L
    if (L[v + 1] != '#') {
        R[v + 1] = 'L';
        f(v + 1, R);
    }

    // D
    if (v + 2 < n && L[v + 2] != '#') {
        R[v + 1] = 'D';
        f(v + 2, R);
    }
    R[v + 1] = '#';
}
}

```

```

void solve(istream& cin, ostream& cout) {
    cin >> n;
    string face;

```

```
cin >> face;
L = string(n, '#');
while (true) {
    string x;
    cin >> x;
    if (x == "END") {
        break;
    }
    int v = atoi(x.c_str());
    int to;
    cin >> to;
    v--, to--;

    if (v == to) {
        L[v] = 'R';
    }
    else if (v < to) {
        L[v] = 'D';
    }
    else {
        L[v] = 'U';
    }
}
for (int v = 0; v < n; v++) {
    if (L[v] != '#') {
```

```
        string R(n, '#');
        f(v, R);
    }
}

    cout << ss.size() << "\n";
}

/*
* TEMPLATE MAIN
*/

int main() {
    //ifstream cin("input.txt");
    ios::sync_with_stdio(false), cout.tie(nullptr), cin.tie(nullptr);

    s64 t = 1;
    //cin >> t;
    while (t--) {
        solve(cin, cout);
    }
    return 0;
}
```

#### Задание 4. Попытка 1.

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
using u8 = uint8_t;
```

```
using u16 = uint16_t;
```

```
using u32 = uint32_t;
```

```
using u64 = uint64_t;
```

```
using s8 = int8_t;
```

```
using s16 = int16_t;
```

```
using s32 = int32_t;
```

```
using s64 = int64_t;
```

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

```
#define all(obj) obj.begin(), obj.end()
```

```
/*
```

```
* SOLVE PROBLEM
```

```
*/
```

```
mt19937 rnd(42);
```

```
struct dsu {
```

```
vector<int> p;
```

```
dsu(int n) {  
    p.resize(n);  
    iota(all(p), 0);  
}
```

```
int find(int v) {  
    return p[v] == v ? v : p[v] = find(p[v]);  
}
```

```
void uni(int a, int b) {  
    a = find(a);  
    b = find(b);  
    if (a != b) {  
        p[a] = b;  
    }  
}  
};
```

```
struct edge {  
    int a, b;  
  
    bool operator < (const edge& rhs) const {  
        if (a == rhs.a) {
```

```

        return b < Rhs.b;
    }
    return a < Rhs.a;
}
};

int n;

vector<vector<int>> g;

vector<edge> ost_e;

// dp[v] = P e P s P » P I P s C T P μ P ± P μ C T P I P i P s P r P r P μ C T P μ P I P μ v P e P · ost_e,
P e P s C, P s C T C < P μ P i P s P » P S P s C T C, C T C T P » P μ P P P C, C, P ° P j

vector<int> dp;

vector<int> cntv;

vector<int> depth, parent, tin, tout;

int timer = 0;

void build(int v, int prev) {
    tin[v] = timer++;
    for (int to : g[v]) {

```

```

    if (to != prev) {
        parent[to] = v;
        depth[to] = depth[v] + 1;
        build(to, v);
    }
}
tout[v] = timer++;
}

```

```

int lca(int a, int b) {
    while (a != b) {
        if (depth[a] < depth[b]) {
            swap(a, b);
        }
        // a = parent[a]
        a = parent[a];
    }
    return a;
}

```

```

bool is_anc(int v, int anc) {
    return tin[anc] <= tin[v] && tin[v] < tout[anc];
}

```

```

void relax_dp(int v, int prev) {

```

```
for (int to : g[v]) {
    if (to != prev) {
        relax_dp(to, v);
        dp[v] += dp[to];
        cntv[v] += cntv[to];
    }
}
}
```

```
void build_dp() {
    {
        timer = 0;
        tin.resize(n), tout.resize(n);
        depth.resize(n), parent.resize(n), cntv.assign(n, 0);
        depth[0] = 0;
        parent[0] = -1;
        build(0, -1);
    }
}
```

```
dp.assign(n, 0);
for (auto [a, b] : ost_e) {
    dp[lca(a, b)]++;
    cntv[a]++;
    cntv[b]++;
}
```



```

    relax_dp(0, -1);
}

int ans, ans_v;

void slv(int v, int prev) {
    // prev->v delete

    // PIPsPrPrPμCτPμPIPμ v PSCfP||PSPs P·PSP°C,CH PεPsP»PIPs CτPμP±PμCτ
    PēP· ost_e

    if (prev != -1) {
        int cntv = cntv[v] - 2 * dp[v] + 1;

        if (ans > cntv) {
            ans = cntv;
            ans_v = v;
        }
    }

    for (int to : g[v]) {
        if (to != prev) {
            slv(to, v);
        }
    }
}

```

```

void solve(istream& cin, ostream& cout) {

    int m;

    cin >> n >> m;

    vector<edge> e(m);

    for (int i = 0; i < m; i++) {

        cin >> e[i].a >> e[i].b;

        e[i].a--, e[i].b--;

    }

    int global_ans = 1e9;

    vector<edge> ans_edge;

    for (int i = 0; i < 100; i++) {

        dsu d(n);

        ost_e.clear();

        g.assign(n, {});

        for (auto [a, b] : e) {

            if (d.find(a) != d.find(b)) {

                g[a].push_back(b);

                g[b].push_back(a);

                d.uni(a, b);

            }

            else {

                ost_e.push_back({ a, b });

            }

        }

    }
}

```

```
}
```

```
build_dp();
```

```
ans = 1e9;
```

```
ans_v = -1;
```

```
slv(0, -1);
```

```
if (global_ans > ans) {
```

```
    global_ans = ans;
```

```
    ans_edge.clear();
```

```
    int v = ans_v;
```

```
    int prev = parent[v];
```

```
    ans_edge.push_back({ v, prev });
```

```
    for (auto [a, b] : ost_e) {
```

```
        if (is_anc(a, v) != is_anc(b, v)) {
```

```
            ans_edge.push_back({ a, b });
```

```
        }
```

```
    }
```

```
}
```

```
shuffle(all(e), rnd);
```

```

    }

    for (auto& [a, b] : ans_edge) {
        if (a > b) {
            swap(a, b);
        }
    }

    sort(all(ans_edge));

    cout << ans_edge.size() << "\n";
    for (auto [a, b] : ans_edge) {
        cout << a + 1 << " " << b + 1 << "\n";
    }
}

/*
* TEMPLATE MAIN
*/

int main() {
    //ifstream cin("input.txt");
    ios::sync_with_stdio(false), cout.tie(nullptr), cin.tie(nullptr);

    s64 t = 1;

```

```
//cin >> t;  
while (t--) {  
    solve(cin, cout);  
}  
return 0;  
}
```

### Задание 5. Попытка 1.

```
#include <bits/stdc++.h>

using namespace std;

using u8 = uint8_t;
using u16 = uint16_t;
using u32 = uint32_t;
using u64 = uint64_t;

using s8 = int8_t;
using s16 = int16_t;
using s32 = int32_t;
using s64 = int64_t;

using ld = long double;

#define all(obj) obj.begin(), obj.end()

/*
 * SOLVE PROBLEM
 */

void solve(istream& cin, ostream& cout) {
    s64 xs, ys, xf, yf;
    cin >> xs >> ys >> xf >> yf;
```

```

set<pair<s64, s64>> vis;

queue<pair<s64, s64>> Q0, Q1;
Q0.push({ xs, ys });
vis.insert({ xs, ys });
s64 d = 1;
while (Q0.size() + Q1.size()) {
    if (Q0.empty()) {
        swap(Q0, Q1);
        d++;
    }

    auto [x, y] = Q0.front();
    Q0.pop();

    if (x == xf && y == yf) {
        cout << d << "\n";
        return;
    }

#define step(nx, ny)\
if (!vis.count({ x + nx, y + ny})) {\
    Q1.push({ x + nx, y + ny });\
    vis.insert({x + nx, y + ny});\
}

```

```
}
```

```
step(0, -1);
```

```
step(0, +1);
```

```
if (abs(x) % 2 == 0) {
```

```
    // green
```

```
    if (abs(y) % 3 == 0) {
```

```
        step(-1, 0);
```

```
        step(+1, 0);
```

```
    }
```

```
    // blue
```

```
    else if ((y % 3 + 3) % 3 == 1) {
```

```
        step(-1, +1);
```

```
    }
```

```
    // red
```

```
    else {
```

```
        step(+1, -1);
```

```
    }
```

```
}
```

```
else {
```

```
    // green
```

```
    if ((y % 3 + 3) % 3 == 1) {
```

```
        step(-1, -1);
```

```
        step(1, 0);
```



```
    }
    else if ((y % 3 + 3) % 3 == 2) {
        step(1, 1);
    }
    // red
    else {
        step(-1, 0);
    }
}
}
}

/*
 * TEMPLATE MAIN
 */

int main() {
    //ifstream cin("input.txt");
    ios::sync_with_stdio(false), cout.tie(nullptr), cin.tie(nullptr);

    s64 t = 1;
    //cin >> t;
    while (t--) {
        solve(cin, cout);
    }
}
```

```
}  
return 0;  
}
```

## Задание 5. Попытка 2.

```
#include <bits/stdc++.h>

using namespace std;

using u8 = uint8_t;
using u16 = uint16_t;
using u32 = uint32_t;
using u64 = uint64_t;

using s8 = int8_t;
using s16 = int16_t;
using s32 = int32_t;
using s64 = int64_t;

using ld = long double;

#define all(obj) obj.begin(), obj.end()

/*
 * SOLVE PROBLEM
 */

void solve(istream& cin, ostream& cout) {
    s64 xs, ys, xf, yf;
    cin >> xs >> ys >> xf >> yf;
```

```

set<pair<s64, s64>> vis;

queue<pair<s64, s64>> Q0, Q1;
Q0.push({ xs, ys });
vis.insert({ xs, ys });
s64 d = 1;
while (Q0.size() + Q1.size()) {
    if (Q0.empty()) {
        swap(Q0, Q1);
        d++;
    }

    auto [x, y] = Q0.front();
    Q0.pop();

    if (x == xf && y == yf) {
        cout << d << "\n";
        return;
    }

#define step(nx, ny)\
if (!vis.count({ x + nx, y + ny})) {\
    Q1.push({ x + nx, y + ny });\
    vis.insert({x + nx, y + ny});\
}

```

```
}
```

```
step(0, -1);
```

```
step(0, +1);
```

```
if (abs(x) % 2 == 0) {
```

```
    // green
```

```
    if (abs(y) % 3 == 0) {
```

```
        step(-1, 0);
```

```
        step(+1, 0);
```

```
    }
```

```
    // blue
```

```
    else if ((y % 3 + 3) % 3 == 1) {
```

```
        step(-1, +1);
```

```
    }
```

```
    // red
```

```
    else {
```

```
        step(+1, -1);
```

```
    }
```

```
}
```

```
else {
```

```
    // green
```

```
    if ((y % 3 + 3) % 3 == 1) {
```

```
        step(-1, 1);
```

```
        step(1, 0);
```

```
    }
    else if ((y % 3 + 3) % 3 == 2) {
        step(1, 1);
    }
    // red
    else {
        step(-1, 0);
    }
}
}
}

/*
 * TEMPLATE MAIN
 */

int main() {
    //ifstream cin("input.txt");
    ios::sync_with_stdio(false), cout.tie(nullptr), cin.tie(nullptr);

    s64 t = 1;
    //cin >> t;
    while (t--) {
        solve(cin, cout);
    }
}
```

```
}  
return 0;  
}
```