



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

ОЛИМПИАДНАЯ РАБОТА

Наименование олимпиады школьников: **«Ломоносов»**

Профиль олимпиады: **Информатика**

ФИО участника олимпиады: **Анисимова Вероника Артёмовна**

Класс: **10 класс**

Технический балл: **88**

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	93	97	0	100	0

Дополнительный балл: 55

Задание 1. Попытка 1.

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

using namespace std;

int k, n;

vector<int> cut(string s) {
    vector<int> ans;
    for (int i = 0; i < s.size(); i++) {
        if (s[i] >= '0' && s[i] <= '9')
            ans.push_back(s[i] - '0');
        else if (s[i] >= 'a' && s[i] <= 'z')
            ans.push_back(s[i] - 'a' + 10);
        else
            ans.push_back(s[i] - 'A' + 36);
    }
    return ans;
}

string cutS(string s) {
    string ans = "";
    int i = 0;
    while (i < s.size() - 1 && s[i] == '0') i++;
    for (; i < s.size(); i++) {
        ans.push_back(s[i]);
    }
}
```

```
    return ans;
}
```

```
bool check(vector<int> &v) {
    if (v.size() == 1 && v[0] == 0) return true;
    for (int i = v.size() - k + 1; i < v.size(); i++)
        if (v[i] != 0) return false;

    if (v[v.size() - k] % k) return false;
    return true;
}
```

```
int main() {
    cin >> k >> n;

    vector<vector<int> > v(n);
    vector<string> vec(n);

    for (int i = 0; i < n; i++) {
        cin >> vec[i];
        vec[i] = cutS(vec[i]);
        v[i] = cut(vec[i]);
    }
```

```
    set<vector<int> > st;
```

```

for (int i = 0; i < n; i++) {
    if (check(v[i])) st.insert(v[i]);
}

if (st.empty()) {
    cout << "-1\n";
    return 0;
}

string s_ans = "";
vector<int> ans = *st.rbegin();

for (int i = 0; i < ans.size(); i++) {
    if (ans[i] >= 0 && ans[i] <= 9)
        s_ans += char('0' + ans[i]);
    else if (ans[i] >= 10 && ans[i] <= 35)
        s_ans += char('a' + ans[i] - 10);
    else
        s_ans += char('A' + ans[i] - 36);
}

cout << s_ans << "\n";

for (int i = 0; i < n; i++) {
    if (vec[i] == s_ans) cout << i + 1 << "\n";
}

```

```
}
```

```
return 0;
```

```
}
```

Задание 2. Попытка 1.

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

using namespace std;

int main() {

    int n;

    cin >> n;

    string s;

    cin >> s;

    multiset<int> ms;

    for (auto i : s) {

        if (i >= '0' && i <= '9') ms.insert(i - '0');

        if (i >= 'a' && i <= 'z') ms.insert(i - 'a' + 10);

        if (i >= 'A' && i <= 'Z') ms.insert(i - 'A' + 36);

    }

    int lenS = s.size(), mx = 0;

    multiset<int> check = ms;

    vector<int> ans(62);

    for (int i = 1; i <= min(61, lenS); i++) {

        if (*check.begin() <= i)

            check.erase(check.begin());

        else

            break;

    }
```

```

    mx = i;
}

if (mx == 0) {
    cout << "-1\n";
    return 0;
}

for (int i = mx; i >= 1; i--) {
    multiset<int>::iterator it = lower_bound(ms.begin(), ms.end(), i);
    if (it == ms.end() || *it > i) it--;
    ans[i] = *it;
    ms.erase(it);
}

bool flag = true;

for (int i = mx; i >= 1; i--) {
    if (i > 1 && ans[i] == 0 && flag) continue;
    flag = false;
    if (ans[i] >= 0 && ans[i] <= 9) {
        cout << ans[i];
    } else if (ans[i] >= 10 && ans[i] <= 35) {
        ans[i] -= 10;
        cout << char('a' + ans[i]);
    }
}

```



```
    } else {  
        ans[i] -= 36;  
        cout << char(ans[i] + 'A');  
    }  
}  
  
return 0;  
}
```

Задание 4. Попытка 1.

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

using namespace std;

#define INF 1000000000

const int MAXN = 105;

int n, g[MAXN][MAXN];

int best_cost = 1000000000;

vector<int> best_cut;

void mincut() {
    vector<int> v[MAXN];

    for (int i = 0; i < n; ++i) v[i].assign(1, i);

    int w[MAXN];

    bool exist[MAXN], in_a[MAXN];

    memset(exist, true, sizeof exist);

    for (int ph = 0; ph < n - 1; ++ph) {
        memset(in_a, false, sizeof in_a);

        memset(w, 0, sizeof w);

        for (int it = 0, prev; it < n - ph; ++it) {
            int sel = -1;

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

                if (exist[i] && !in_a[i] && (sel == -1 || w[i] > w[sel])) sel = i;
```

```

if (it == n - ph - 1) {
    if (w[sel] < best_cost) best_cost = w[sel], best_cut = v[sel];
    v[prev].insert(v[prev].end(), v[sel].begin(), v[sel].end());
    for (int i = 0; i < n; ++i) g[prev][i] = g[i][prev] += g[sel][i];
    exist[sel] = false;
} else {
    in_a[sel] = true;
    for (int i = 0; i < n; ++i) w[i] += g[sel][i];
    prev = sel;
}
}
}
}
}

```

```

int main() {
    int m, x, y;
    cin >> n >> m;
    vector<vector<int>> G(n + 1, vector<int>());
    while (m--) {
        int x, y;
        cin >> x >> y;
        G[y].push_back(x);
        G[x].push_back(y);
        x--, y--;
        g[x][y]++, g[y][x]++;
    }
}

```

```
}

mincut();

cout << best_cost << "\n";

set<int> best_set;

vector<pair<int, int> > ans;

for (auto &i : best_cut) best_set.insert(i + 1);

for (auto i : best_set) {
    for (auto j : G[i])
        if (best_set.find(j) == best_set.end())
            ans.push_back({min(i, j), max(i, j)});
}

sort(ans.begin(), ans.end());

for (auto i : ans) cout << i.first << ' ' << i.second << "\n";

return 0;

}
```