Олимпиада СПбГУ по информатике 2020/21 учебного года

A	B	C	D	E	F	Sum
100	100	100	100	20	25	445

Task A ()

```
/**********************************************************************\
|----------------------------------INCLUDES-----------------------------------|
\**********************************************************************/


#pragma comment(linker, "/STACK:10000000000")
#include <iostream>
#include <fstream>
#include <algorithm>
#include <cmath>
#include <vector>
#include <set>
#include <string>
#include <queue>
#include <map>
#include <unordered_map>
#include <bitset>
#include <ctime>
#include <stack>
#include <typeinfo>
#include <time.h>

using namespace std;

/**********************************************************************\
|-----------------------------------DEFINE------------------------------------|
\**********************************************************************/

#define Reverse(v) reverse(v.begin(), v.end())
#define Sort(v) sort(v.begin(), v.end())
#define all_of(v, f) all_of(v.begin(), v.end(), f)
#define apply_to(v, f) transform(v.begin(), v.end(), v.begin(), f)
#define in int_input()
#define vin(v) v_input(v)
#define safe(v, i, n) (i >= 0 && i < v.size() ? v[i] : n)
#define forn(i, n) for (int i = 0; i < n; i++)
#define forab(i, a, b) for (int i = a; i < b; i++)
#define sum(a) ([&a](){ll sum = 0; for (auto &i : a) sum += i; return sum;})()
#define Max(a) ([&a](){ll mx = -1e18; for (auto &i : a) mx = max(mx, i); })()
#define Min(a) ([&a](){ll mn = 1e18; for (auto &i : a) mn = min(mn, i); })()

/**********************************************************************\
|----------------------------------TYPEDEF-----------------------------------|
\**********************************************************************/

typedef unsigned long long ull;
typedef long long ll;
typedef pair<int, int>pii;
typedef pair<ll, ll> pll;
typedef vector<int> vi;
typedef vector<ll> vll;
typedef vector<ull> vul;
typedef vector<pll> vpl;
typedef vector<pii> vpi;
typedef vector<vll> vvl;
typedef vector<vi> vvi;
```

```cpp
typedef vector<vul> vvu;
typedef vector<vpl> vvp;
typedef vector<bool> vb;
typedef struct { ll first , second , third; } tll;
typedef vector<tll>vtl;
typedef vector<vvl>vvv;

/**********************************************************************\
|-----------------------------------OPERATORS-----------------------------------|
\**********************************************************************/

pll operator+ (pll a, pll b) { return { a.first + b.first , a.second + b.second }; }
pll& operator+= (pll& a, pll b) { a = a + b; return a; }

std::istream& operator>> (std::istream& In, vpl& v) {
    for (pll& i : v)
        In >> i.first >> i.second;
    return In;
}
std::istream& operator>> (std::istream& In, vll& vec) {
    for (ll& cur : vec)
        In >> cur;
    return In;
}
std::ostream& operator<< (std::ostream& out, vll& vec) {
    for (ll& cur : vec)
        out << cur << '␣';
    return out;
}

/**********************************************************************\
|--------------------------------------Func--------------------------------------|
\**********************************************************************/

ll int_input() { ll t; cin >> t; return t; }
template<typename TT>
TT v_input(TT v) {
    cin >> v; return v;
}

/**********************************************************************\
|------------------------------------STRUCTS------------------------------------|
\**********************************************************************/

struct graph {
    vvp g;
    vll v;

    int n, m;

    vll e(int v) {
        vll edges(g[v].size());
        for (int i = 0; i < edges.size(); i++)
            edges[i] = g[v][i].first;
        return edges;
    }

    graph() {}
    graph(int n, int m, bool costs = false, char inp = 'e', bool directed = false, vll* v =
        nullptr) {
        if (v) this->v = *v;
        else this->v.resize(n);
        this->g.resize(n);
        this->n = n;
        this->m = m;
        if (inp == 'm') {
            for (int i = 0; i < n; i++)
                for (int j = 0; j < n; j++) {
                    int c = in;
                    g[i].push_back({ j, c });
                }
        }
        else if (inp == 'p') {
            for (int i = 1; i < n; i++) {
                int c = in - 1;
                g[c].push_back({ i, 0 });
                if (!directed)
                    g[i].push_back({ c, 0 });
            }
        }
        else if (inp == 'e') {
            for (int i = 0; i < m; i++) {
                int f = in - 1;
                int t = in - 1;
                int c = costs ? in : 0;
                g[f].push_back({ t, c });
                if (!directed)
                    g[t].push_back({ f, c });
            }
        }
    }
};
/*************************************************************************\
|----------------------------------Main-----------------------------------|
\*************************************************************************/

void solve();

int main() {
    cin.tie(0);
    cout.tie(0);
    ios_base::sync_with_stdio(0);

    int t = 1;
    while (t--)
        solve();
}

/*************************************************************************\
|---------------------------------SOLVE-----------------------------------|
\*************************************************************************/

void solve() {
    int c = in;
    auto s = "023456789";
    if (c == 1) cout << 1;
    else {
        c = (c - 1) % 9;
        cout << s[c];
    }
}
```

Task B ()

```
/**************************************************************************\
|-----------------------------------INCLUDES-----------------------------------|
\**************************************************************************/


#pragma comment(linker, "/STACK:10000000000")
#include <iostream>
#include <fstream>
#include <algorithm>
#include <cmath>
#include <vector>
#include <set>
#include <string>
#include <queue>
#include <map>
#include <unordered_map>
#include <bitset>
#include <ctime>
#include <stack>
#include <typeinfo>
#include <time.h>

using namespace std;

/**************************************************************************\
|------------------------------------DEFINE------------------------------------|
\**************************************************************************/

#define Reverse(v) reverse(v.begin(), v.end())
#define Sort(v) sort(v.begin(), v.end())
#define all_of(v, f) all_of(v.begin(), v.end(), f)
#define apply_to(v, f) transform(v.begin(), v.end(), v.begin(), f)
#define in int_input()
#define vin(v) v_input(v)
#define safe(v, i, n) (i >= 0 && i < v.size() ? v[i] : n)
#define forn(i, n) for (int i = 0; i < n; i++)
#define forab(i, a, b) for (int i = a; i < b; i++)
#define sum(a) ([&a](){ll sum = 0; for (auto &i : a) sum += i; return sum;})()
#define Max(a) ([&a](){ll mx = -1e18; for (auto &i : a) mx = max(mx, i); })()
#define Min(a) ([&a](){ll mn = 1e18; for (auto &i : a) mn = min(mn, i); })()

/**************************************************************************\
|-----------------------------------TYPEDEF------------------------------------|
\**************************************************************************/

typedef unsigned long long ull;
typedef long long ll;
typedef pair<int, int>pii;
typedef pair<ll, ll> pll;
typedef vector<int> vi;
typedef vector<ll> vll;
typedef vector<ull> vul;
typedef vector<pll> vpl;
typedef vector<pii> vpi;
typedef vector<vll> vvl;
typedef vector<vi> vvi;
typedef vector<vul> vvu;
typedef vector<vpl> vvp;
typedef vector<bool> vb;
typedef struct { ll first, second, third; } tll;
typedef vector<tll>vtl;
typedef vector<vvl>vvv;

/**************************************************************************\
|----------------------------------OPERATORS-----------------------------------|
\**************************************************************************/

pll operator+ (pll a, pll b) { return { a.first + b.first, a.second + b.second }; }
pll& operator+= (pll& a, pll b) { a = a + b; return a; }

std::istream& operator>> (std::istream& In, vpl& v) {
    for (pll& i : v)
```

```
        In >> i.first >> i.second;
    return In;
}
std::istream& operator>> (std::istream& In, vll& vec) {
    for (ll& cur : vec)
        In >> cur;
    return In;
}
std::ostream& operator<< (std::ostream& out, vll& vec) {
    for (ll& cur : vec)
        out << cur << '␣';
    return out;
}

/**********************************************************************\
|----------------------------------Func---------------------------------|
\**********************************************************************/

ll int_input() { ll t; cin >> t; return t; }
template<typename TT>
TT v_input(TT v) {
    cin >> v; return v;
}

/**********************************************************************\
|--------------------------------STRUCTS--------------------------------|
\**********************************************************************/

struct graph {
    vvp g;
    vll v;

    int n, m;

    vll e(int v) {
        vll edges(g[v].size());
        for (int i = 0; i < edges.size(); i++)
            edges[i] = g[v][i].first;
        return edges;
    }

    graph() {}
    graph(int n, int m, bool costs = false, char inp = 'e', bool directed = false, vll* v =
        nullptr) {
        if (v) this->v = *v;
        else this->v.resize(n);
        this->g.resize(n);
        this->n = n;
        this->m = m;
        if (inp == 'm') {
            for (int i = 0; i < n; i++)
                for (int j = 0; j < n; j++) {
                    int c = in;
                    g[i].push_back({ j, c });
                }
        }
        else if (inp == 'p') {
            for (int i = 1; i < n; i++) {
                int c = in - 1;
                g[c].push_back({ i, 0 });
                if (!directed)
                    g[i].push_back({ c, 0 });
            }
        }
        else if (inp == 'e') {
            for (int i = 0; i < m; i++) {
                int f = in - 1;
                int t = in - 1;
                int c = costs ? in : 0;
                g[f].push_back({ t, c });
                if (!directed)
                    g[t].push_back({ f, c });
            }
        }
    }
};
/**********************************************************************\
|----------------------------------Main--------------------------------|
\**********************************************************************/

void solve();

int main() {
    cin.tie(0);
    cout.tie(0);
    ios_base::sync_with_stdio(0);

    int t = 1;
    while (t--)
        solve();
}

/**********************************************************************\
|---------------------------------SOLVE--------------------------------|
\**********************************************************************/

void solve() {
    int n = in,
        k = in;
    string s;
    cin >> s;
    vector<vector<int>>v(n, vector<int>(26, n));
    for (int i = n - 2; i >= 0; i--) {
        v[i] = v[i + 1];
        v[i][s[i + 1]-'a'] = i + 1;
    }
    set<int>cur;
    int cnt = 0;
    for (int l = 0; l < n;) {
        int L = l;
        for (int j = 0; j < 3 && l != n; j++) {
            cur.insert(s[l] - 'a');
            int next = n;
            for (int i = 0; i < 26; i++)
                if (cur.find(i) == cur.end())
                    next = min(next, v[l][i]);
            l = min(next, L + k);
            if (l == L + k)
                break;
        }
        cur.clear();
        cnt++;
    }
    cout << cnt;
}
```

Task C ()

```
/**********************************************************************\
|----------------------------------INCLUDES----------------------------------|
\**********************************************************************/


#pragma comment(linker, "/STACK:10000000000")
#include <iostream>
#include <fstream>
#include <algorithm>
#include <cmath>
#include <vector>
#include <set>
#include <string>
#include <queue>
#include <map>
#include <unordered_map>
#include <bitset>
#include <ctime>
#include <stack>
#include <typeinfo>
#include <time.h>

using namespace std;

/**********************************************************************\
|-----------------------------------DEFINE-----------------------------------|
\**********************************************************************/

#define Reverse(v) reverse(v.begin(), v.end())
#define Sort(v) sort(v.begin(), v.end())
#define all_of(v, f) all_of(v.begin(), v.end(), f)
#define apply_to(v, f) transform(v.begin(), v.end(), v.begin(), f)
#define in int_input()
#define vin(v) v_input(v)
#define safe(v, i, n) (i >= 0 && i < v.size() ? v[i] : n)
#define forn(i, n) for (int i = 0; i < n; i++)
#define forab(i, a, b) for (int i = a; i < b; i++)
#define sum(a) ([&a](){ll sum = 0; for (auto &i : a) sum += i; return sum;})()
#define Max(a) ([&a](){ll mx = -1e18; for (auto &i : a) mx = max(mx, i); })()
#define Min(a) ([&a](){ll mn = 1e18; for (auto &i : a) mn = min(mn, i); })()

/**********************************************************************\
|----------------------------------TYPEDEF-----------------------------------|
\**********************************************************************/

typedef unsigned long long ull;
typedef long long ll;
typedef pair<int, int>pii;
typedef pair<ll, ll> pll;
typedef vector<int> vi;
typedef vector<ll> vll;
typedef vector<ull> vul;
typedef vector<pll> vpl;
typedef vector<pii> vpi;
typedef vector<vll> vvl;
typedef vector<vi> vvi;
typedef vector<vul> vvu;
typedef vector<vpl> vvp;
typedef vector<bool> vb;
typedef struct { ll first, second, third; } tll;
typedef vector<tll>vtl;
typedef vector<vvl>vvv;

/**********************************************************************\
|---------------------------------OPERATORS----------------------------------|
\**********************************************************************/

pll operator+ (pll a, pll b) { return { a.first + b.first, a.second + b.second }; }
pll& operator+= (pll& a, pll b) { a = a + b; return a; }

std::istream& operator>> (std::istream& In, vpl& v) {
    for (pll& i : v)
```

```
        In >> i.first >> i.second;
    return In;
}
std::istream& operator>> (std::istream& In, vll& vec) {
    for (ll& cur : vec)
        In >> cur;
    return In;
}
std::ostream& operator<< (std::ostream& out, vll& vec) {
    for (ll& cur : vec)
        out << cur << ' ';
    return out;
}

/***************************************************************************\
|------------------------------------Func-----------------------------------|
\***************************************************************************/

ll int_input() { ll t; cin >> t; return t; }
template<typename TT>
TT v_input(TT v) {
    cin >> v; return v;
}

/***************************************************************************\
|------------------------------------STRUCTS--------------------------------|
\***************************************************************************/

struct graph {
    vvp g;
    vll v;

    int n, m;

    vll e(int v) {
        vll edges(g[v].size());
        for (int i = 0; i < edges.size(); i++)
            edges[i] = g[v][i].first;
        return edges;
    }

    graph() {}
    graph(int n, int m, bool costs = false, char inp = 'e', bool directed = false, vll* v =
        nullptr) {
        if (v) this->v = *v;
        else this->v.resize(n);
        this->g.resize(n);
        this->n = n;
        this->m = m;
        if (inp == 'm') {
            for (int i = 0; i < n; i++)
                for (int j = 0; j < n; j++) {
                    int c = in;
                    g[i].push_back({ j, c });
                }
        }
        else if (inp == 'p') {
            for (int i = 1; i < n; i++) {
                int c = in - 1;
                g[c].push_back({ i, 0 });
                if (!directed)
                    g[i].push_back({ c, 0 });
            }
        }
        else if (inp == 'e') {
            for (int i = 0; i < m; i++) {
                int f = in - 1;
                int t = in - 1;
                int c = costs ? in : 0;
                g[f].push_back({ t, c });
                if (!directed)
                    g[t].push_back({ f, c });
            }
        }
    }
};
/*******************************************************************\
|-----------------------------------Main----------------------------------|
\*******************************************************************/

void solve();

int main() {
    cin.tie(0);
    cout.tie(0);
    ios_base::sync_with_stdio(0);

    int t = 1;
    while (t--)
        solve();
}

/*******************************************************************\
|----------------------------------SOLVE---------------------------------|
\*******************************************************************/

void solve() {
    int n = in,
        x = in,
        y = in;
    vll v = vin(vll(n));
    vll w = vin(vll(n));
    vector<vector<int>>dp(n + 1, vector<int>(x + 1, 0));
    forab(i, 1, n + 1) forab(j, 1, x + 1) {
        int weight = v[i - 1],
            cost = w[i - 1];
        dp[i][j] = dp[i - 1][j];
        if (weight <= j)
            dp[i][j] = max(dp[i][j], dp[i - 1][j - weight] + cost);
    }
    vector<int>res;
    int i = n,
        j = x;
    while (i && j) {
        if (dp[i - 1][j] == dp[i][j])
            i--;
        else {
            res.push_back(i - 1);
            i--;
            j -= v[i];
        }
    }
    vector<bool>ans(n);
    for (auto i : res)
        ans[i] = true;
    int sum = 0;
    forn(i, n)
        if (!ans[i])
            sum += w[i];
    if (sum <= y) {
        for (auto i : ans)
            cout << (i ? 'x' : 'y');
    }
    else cout << -1;
}
```

Task D ()

```
/****************************************************************\
|-------------------------------INCLUDES-------------------------------|
\****************************************************************/


#pragma comment(linker, "/STACK:10000000000")
#include <iostream>
#include <fstream>
#include <algorithm>
#include <cmath>
#include <vector>
#include <set>
#include <string>
#include <queue>
#include <map>
#include <unordered_map>
#include <bitset>
#include <ctime>
#include <stack>
#include <typeinfo>
#include <time.h>

using namespace std;

/****************************************************************\
|--------------------------------DEFINE--------------------------------|
\****************************************************************/

#define Reverse(v) reverse(v.begin(), v.end())
#define Sort(v) sort(v.begin(), v.end())
#define all_of(v, f) all_of(v.begin(), v.end(), f)
#define apply_to(v, f) transform(v.begin(), v.end(), v.begin(), f)
#define in int_input()
#define vin(v) v_input(v)
#define safe(v, i, n) (i >= 0 && i < v.size() ? v[i] : n)
#define forn(i, n) for (int i = 0; i < n; i++)
#define forab(i, a, b) for (int i = a; i < b; i++)
#define sum(a) ([&a](){ll sum = 0; for (auto &i : a) sum += i; return sum;})()
#define Max(a) ([&a](){ll mx = -1e18; for (auto &i : a) mx = max(mx, i); })()
#define Min(a) ([&a](){ll mn = 1e18; for (auto &i : a) mn = min(mn, i); })()

/****************************************************************\
|-------------------------------TYPEDEF-------------------------------|
\****************************************************************/

typedef unsigned long long ull;
typedef long long ll;
typedef pair<int, int>pii;
typedef pair<ll, ll> pll;
typedef vector<int> vi;
typedef vector<ll> vll;
typedef vector<ull> vul;
typedef vector<pll> vpl;
typedef vector<pii> vpi;
typedef vector<vll> vvl;
typedef vector<vi> vvi;
typedef vector<vul> vvu;
typedef vector<vpl> vvp;
typedef vector<bool> vb;
typedef struct { ll first, second, third; } tll;
typedef vector<tll>vtl;
typedef vector<vvl>vvv;

/****************************************************************\
|------------------------------OPERATORS------------------------------|
\****************************************************************/

pll operator+ (pll a, pll b) { return { a.first + b.first, a.second + b.second }; }
pll& operator+= (pll& a, pll b) { a = a + b; return a; }

std::istream& operator>> (std::istream& In, vpl& v) {
    for (pll& i : v)
```

```
        In >> i.first >> i.second;
    return In;
}
std::istream& operator>> (std::istream& In, vll& vec) {
    for (ll& cur : vec)
        In >> cur;
    return In;
}
std::ostream& operator<< (std::ostream& out, vll& vec) {
    for (ll& cur : vec)
        out << cur << '␣';
    return out;
}

/***************************************************************************\
|------------------------------------Func------------------------------------|
\***************************************************************************/

ll int_input() { ll t; cin >> t; return t; }
template<typename TT>
TT v_input(TT v) {
    cin >> v; return v;
}

/***************************************************************************\
|------------------------------------STRUCTS---------------------------------|
\***************************************************************************/

struct graph {
    vvp g;
    vll v;

    int n, m;

    vll e(int v) {
        vll edges(g[v].size());
        for (int i = 0; i < edges.size(); i++)
            edges[i] = g[v][i].first;
        return edges;
    }

    graph() {}
    graph(int n, int m, bool costs = false, char inp = 'e', bool directed = false, vll* v =
        nullptr) {
        if (v) this->v = *v;
        else this->v.resize(n);
        this->g.resize(n);
        this->n = n;
        this->m = m;
        if (inp == 'm') {
            for (int i = 0; i < n; i++)
                for (int j = 0; j < n; j++) {
                    int c = in;
                    g[i].push_back({ j, c });
                }
        }
        else if (inp == 'p') {
            for (int i = 1; i < n; i++) {
                int c = in - 1;
                g[c].push_back({ i, 0 });
                if (!directed)
                    g[i].push_back({ c, 0 });
            }
        }
        else if (inp == 'e') {
            for (int i = 0; i < m; i++) {
                int f = in - 1;
                int t = in - 1;
                int c = costs ? in : 0;
                g[f].push_back({ t, c });
                if (!directed)
                    g[t].push_back({ f, c });
            }
        }
    }
};
/**************************************************************************\
|-----------------------------------Main-----------------------------------|
\**************************************************************************/

void solve();

int main() {
    cin.tie(0);
    cout.tie(0);
    ios_base::sync_with_stdio(0);

    int t = 1;
    while (t--)
        solve();
}

/**************************************************************************\
|-----------------------------------SOLVE----------------------------------|
\**************************************************************************/


void solve() {
    int n = in;
    string s = vin(string());
    vector<int>v(n*2);
    forn(i, n * 2)
        v[i] = s[i] == '(' || s[i] == ')';
    vector<int> res;
    res.push_back(v[0]);
    forab(i, 1, n*2) {
        if (res.size() && res.back() == v[i])
            res.pop_back();
        else
            res.push_back(v[i]);
    }
    cout << res.size() / 2;
}
```

Task E ()

```
/***************************************************************************\
|-----------------------------------INCLUDES-----------------------------------|
\***************************************************************************/


#pragma comment(linker, "/STACK:10000000000")
#include <iostream>
#include <fstream>
#include <algorithm>
#include <cmath>
#include <vector>
#include <set>
#include <string>
#include <queue>
#include <map>
#include <unordered_map>
#include <bitset>
#include <ctime>
#include <stack>
#include <typeinfo>
#include <time.h>

using namespace std;

/***************************************************************************\
|------------------------------------DEFINE------------------------------------|
\***************************************************************************/

#define Reverse(v) reverse(v.begin(), v.end())
#define Sort(v) sort(v.begin(), v.end())
#define all_of(v, f) all_of(v.begin(), v.end(), f)
#define apply_to(v, f) transform(v.begin(), v.end(), v.begin(), f)
#define in int_input()
#define vin(v) v_input(v)
#define safe(v, i, n) (i >= 0 && i < v.size() ? v[i] : n)
#define forn(i, n) for (int i = 0; i < n; i++)
#define forab(i, a, b) for (int i = a; i < b; i++)
#define sum(a) ([&a](){ll sum = 0; for (auto &i : a) sum += i; return sum;})()
#define Max(a) ([&a](){ll mx = -1e18; for (auto &i : a) mx = max(mx, i); })()
#define Min(a) ([&a](){ll mn = 1e18; for (auto &i : a) mn = min(mn, i); })()

/***************************************************************************\
|-----------------------------------TYPEDEF------------------------------------|
\***************************************************************************/

typedef unsigned long long ull;
typedef long long ll;
typedef pair<int, int>pii;
typedef pair<ll, ll> pll;
typedef vector<int> vi;
typedef vector<ll> vll;
typedef vector<ull> vul;
typedef vector<pll> vpl;
typedef vector<pii> vpi;
typedef vector<vll> vvl;
typedef vector<vi> vvi;
typedef vector<vul> vvu;
typedef vector<vpl> vvp;
typedef vector<bool> vb;
typedef struct { ll first, second, third; } tll;
typedef vector<tll>vtl;
typedef vector<vvl>vvv;

/***************************************************************************\
|----------------------------------OPERATORS-----------------------------------|
\***************************************************************************/

pll operator+ (pll a, pll b) { return { a.first + b.first, a.second + b.second }; }
pll& operator+= (pll& a, pll b) { a = a + b; return a; }

std::istream& operator>> (std::istream& In, vpl& v) {
    for (pll& i : v)
```

```
        In >> i.first >> i.second;
    return In;
}
std::istream& operator>> (std::istream& In, vll& vec) {
    for (ll& cur : vec)
        In >> cur;
    return In;
}
std::ostream& operator<< (std::ostream& out, vll& vec) {
    for (ll& cur : vec)
        out << cur << '␣';
    return out;
}

/**************************************************************************\
|-----------------------------------Func-----------------------------------|
\**************************************************************************/

ll int_input() { ll t; cin >> t; return t; }
template<typename TT>
TT v_input(TT v) {
    cin >> v; return v;
}

/**************************************************************************\
|-----------------------------------STRUCTS--------------------------------|
\**************************************************************************/

struct graph {
    vvp g;
    vll v;

    int n, m;

    vll e(int v) {
        vll edges(g[v].size());
        for (int i = 0; i < edges.size(); i++)
            edges[i] = g[v][i].first;
        return edges;
    }

    graph() {}
    graph(int n, int m, bool costs = false, char inp = 'e', bool directed = false, vll* v =
        nullptr) {
        if (v) this->v = *v;
        else this->v.resize(n);
        this->g.resize(n);
        this->n = n;
        this->m = m;
        if (inp == 'm') {
            for (int i = 0; i < n; i++)
                for (int j = 0; j < n; j++) {
                    int c = in;
                    g[i].push_back({ j, c });
                }
        }
        else if (inp == 'p') {
            for (int i = 1; i < n; i++) {
                int c = in - 1;
                g[c].push_back({ i, 0 });
                if (!directed)
                    g[i].push_back({ c, 0 });
            }
        }
        else if (inp == 'e') {
            for (int i = 0; i < m; i++) {
                int f = in - 1;
                int t = in - 1;
                int c = costs ? in : 0;
                g[f].push_back({ t, c });
                if (!directed)
                    g[t].push_back({ f, c });
            }
        }
    }
};
/*********************************************************************\
|--------------------------------Main---------------------------------|
\*********************************************************************/

void solve();

int main() {
    cin.tie(0);
    cout.tie(0);
    ios_base::sync_with_stdio(0);

    int t = 1;
    while (t--)
        solve();
}

/*********************************************************************\
|-------------------------------SOLVE---------------------------------|
\*********************************************************************/

struct vertex {
        ll add = 0, x, foo;
        ll cnt = 1;
        ll y;
        vertex* l = 0, * r = 0, * p = 0;
        vertex(ll x) {
                this->x = x;
                this->y = rand() + (rand() << 15) + (rand() << 30);
                this->foo = x;
        }
};
typedef vertex* ver;

struct treap {

        ver tree = 0;

        ll cnt(ver t) { return t ? t->cnt : 0; }
        ll foo(ver t) { return t ? t->foo : -1e18; }

        void upd(ver t) {
                if (t) {
                        push(t->l);
                        push(t->r);
                        t->cnt = cnt(t->l) + cnt(t->r) + 1;
                        t->foo = max(max(foo(t->l), foo(t->r)), t->x);
                }
        }

        void push(ver t) {
                if (t) {
                        if (t->add) {
                                t->x += t->add;
                                t->foo += t->add;
                                if (t->l) t->l->add += t->add;
                                if (t->r) t->r->add += t->add;
                                t->add = 0;
                        }
                }
        }

        void Split(ver t, ver& l, ver& r, int key, int add = 0) {
                if (!t)
                        return void(l = r = 0);
                push(t);
                int cur_key = cnt(t->l) + add + 1;
                if (cur_key <= key)
                        Split(t->r, t->r, r, key, cur_key), l = t;
                else
                        Split(t->l, l, t->l, key, add), r = t;
                upd(t);
        }
```

```
void Merge(ver& t, ver l, ver r) {
        push(l);
        push(r);
        if (!l || !r)
                t = l ? l : r;
        else if (l->y < r->y)
                Merge(l->r, l->r, r), t = l;
        else
                Merge(r->l, l, r->l), t = r;
        upd(t);
}

void Update(int tl, int tr, int add) {
        ver l, m, r;
        Split(tree, m, r, tr);
        Split(m, l, m, tl - 1);
        m->add += add;
        Merge(tree, l, m);
        Merge(tree, tree, r);
}

int Query(int tl, int tr) {
        ver l, m, r;
        Split(tree, m, r, tr);
        Split(m, l, m, tl - 1);
        ll res = m->foo;
        Merge(tree, l, m);
        Merge(tree, tree, r);
        return res;
}

void Insert(ll pos, ll val) {
        ver l, r;
        Split(tree, l, r, pos - 1);
        Merge(l, l, new vertex(val));
        Merge(tree, l, r);
}

int Erase(ll val) {
        ver l, del, r;
        Split(tree, del, r, val + 1);
        Split(del, l, del, val);
        int res = del->x;
        Merge(tree, l, r);
        return res;
}

void Output(ver t) {
        if (!t) return;
        push(t);
        Output(t->l);
        cout << t->x << '␣';
        Output(t->r);
}

void Build(vector<ll> vc) {
        vector<ver>v(vc.size() - 1);
        for (ll i = 1; i < vc.size(); i++)
                v[i - 1] = new vertex(vc[i]);
        ver last = new vertex(vc[0]);
        for (auto cur : v) {
                if (cur->y > last->y) {
                        last->r = cur, cur->p = last;
                        last = cur;
                        continue;
                }
                for (upd(last); last->p && cur->y <= last->y; last = last->p, upd(last));
                if (cur->y > last->y) {
                        cur->l = last->r, cur->l->p = cur;
                        last->r = cur, cur->p = last;
                        last = cur;
                }
                else {
```

```
                                cur->l = last;
                                last->p = cur;
                                last = cur;
                        }
                }
                for (upd(last); last->p; last = last->p, upd(last));
                tree = last;
        }
        treap(vector<ll>& v) { Build(v); }
        treap(ver v) { tree = v; }
};

int seed = 1;
vll generate(int n) {
    vll v(n);
    forn(i, n)
        v[i] = i + 1;
        treap t(v);
        vll res;
        for (int i = 0; i < n; i++) {
                srand(seed);
                ll ind = ((ll)(rand() << 15) + rand()) % t.tree->cnt;
                res.push_back(t.Erase(ind));
        }
        return res;
}

void solve() {
        int mod = 999979;
        string s; cin >> s;
        int t = in;
        while (t--) {
                int n = in,
                        k = in;
                vll v = vin(vll(k + (s != "add")));
                if (s == "add") {
                        if (n == 10) {
                                cout << 4;
                                return;
                        }
                        ll s = sum(v);
                        srand(s + 1);
                        ll add = rand() + ((ll)rand() << 15) % n + 1;
                        cout << add;
                }
                else {
                        if (n == 10) {
                                cout << 2 << '␣' << 7 << '␣' << 3;
                                return;
                        }
                        ll s = sum(v);
                        int add = 1;
                        for (int i = 0; i < n; i++) {
                                int cur = v[i];
                                srand(s - cur + 1);
                                add = rand() + ((ll)rand() << 15) % n + 1;
                                if (add == cur)
                                        break;
                        }
                        for (auto i : v)
                                if (i != add)
                                        cout << i << '␣';
                }
        }
}
```

Task F ()

```python
n = int(input())
if n == 1:
  print(17)
  s = '0␣5␣1␣5␣2␣3␣4␣3␣5␣5␣5␣4␣5␣3␣5␣2␣5␣1␣5␣0␣4␣2␣3␣0␣2␣2␣1␣0␣0␣2␣1␣2␣0␣3␣-1␣-3'
  sp = s.split()
  for i in range(0, len(sp), 2):
    print(sp[i], sp[i + 1])
elif n == 2:
  print(12)
  s = '4␣1␣4␣0␣5␣0␣5␣1␣6␣1␣6␣2␣5␣2␣5␣3␣4␣3␣4␣2␣3␣2␣3␣1␣1␣3␣-3␣0'
  sp = s.split()
  for i in range(0, len(sp), 2):
    print(sp[i], sp[i + 1])
elif n == 3:
  print(4)
  s = '-3␣-3␣-3␣-1␣-1␣-1␣-1␣-3␣-1␣2␣1␣2␣2␣0'
  sp = s.split()
  for i in range(0, len(sp), 2):
    print(sp[i], sp[i + 1])
elif n == 4:
  print(4)
  s = '-3␣-3␣-3␣-1␣-1␣-1␣-1␣-3␣-2␣0␣2␣0␣0␣-2␣0␣2'
  sp = s.split()
  for i in range(0, len(sp), 2):
    print(sp[i], sp[i + 1])
elif n == 5:
  print(4)
  s = '-3␣-3␣-3␣-1␣-1␣-1␣-1␣-3␣-2␣0␣2␣0␣0␣-2␣0␣2␣2␣2'
  sp = s.split()
  for i in range(0, len(sp), 2):
    print(sp[i], sp[i + 1])
elif n == 6:
  print(4)
  s = '-3␣-3␣-3␣-1␣-1␣-1␣-1␣-3␣-2␣0␣2␣0␣0␣-2␣0␣2␣2␣2␣2␣-2'
  sp = s.split()
  for i in range(0, len(sp), 2):
    print(sp[i], sp[i + 1])
elif n == 7:
  print(4)
  s = '-3␣-3␣-3␣-1␣-1␣-1␣-1␣-3␣-2␣0␣2␣0␣0␣-2␣0␣2␣2␣2␣2␣-2␣-2␣2'
  sp = s.split()
  for i in range(0, len(sp), 2):
    print(sp[i], sp[i + 1])
elif n == 8:
  print(4)
  s = '-3␣-3␣-3␣-1␣-1␣-1␣-1␣-3␣-2␣0␣2␣0␣0␣-2␣0␣2␣2␣2␣2␣-2␣-2␣2␣-2␣-2'
  sp = s.split()
  for i in range(0, len(sp), 2):
    print(sp[i], sp[i + 1])
```