

# Олимпиада СПбГУ по информатике 2021/22 учебного года

A	B	C	D	E	F	Sum
100	100	100	100	58	31	489

## Task A ()

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <set>
#include <unordered_set>
#include <map>
#include <unordered_map>
#include <random>
#include <assert.h>

using namespace std;

typedef long long int ll;
typedef long double ld;

#define print(a) do { cerr << #a << ":\n"; for (auto& e : a) cerr << e << "\n"; cerr << "\n"; } while (0);
#define input(a) do { for (auto& e : a) cin >> e; } while (0);

mt19937_64 rnd(1231);
const int MAXINT = 1e9 + 12;
const ll MAXLL = (ll)1e18 + 21;

pair<int, ll> gettrue(ll x) {
    int cntzero = 0;
    while (x % 10 == 0) {
        cntzero++;
        x /= 10;
    }
    return {cntzero, x};
}

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

    int n;
    cin >> n;
    map<int, ll> values;
    set<int> coolb;
    for (int i = 0; i < n; i++) {
        int a, b;
        cin >> a >> b;
        values[b] += a;
        coolb.insert(b);
    }

    while (true) {
        auto now = *coolb.begin();
        coolb.erase(coolb.begin());
        auto rev = gettrue(values[now]);
        if (rev.first == 0) {
            cout << now << "\n";
            return 0;
        }
        else {
```

```

        coolb.insert(now + rev.first);
        values[now + rev.first] += rev.second;
    }

}

```

## Task B ()

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <set>
#include <unordered_set>
#include <map>
#include <unordered_map>
#include <random>
#include <assert.h>

using namespace std;

typedef long long int ll;
typedef long double ld;

#define print(a) do { cerr << #a << ":_"; for (auto& e : a) cerr << e << " "; cerr << "\n"; }
while (0);
#define input(a) do { for (auto& e : a) cin >> e; } while (0);

mt19937_64 rnd(1231);
const int MAXINT = 1e9 + 12;
const ll MAXLL = (1ll)1e18 + 21;

int read() {
    string s;
    cin >> s;
    if (s == "Burn" || s == "Tired" || s == "Fail")
        exit(0);
    int cnte = 0;
    for (auto& e : s)
        if (e == 'e')
            cnte++;
    return cnte / 2;
}

void wait() {
    cout << "Wait" << endl;
}
void flip() {
    cout << "Flip_and_wait" << endl;
}
void end() {
    cout << "Stop" << endl;
    exit(0);
}

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

    int n;
    cin >> n;
    vector<int> ord(n);
    input(ord);

    flip();
    int cntleft = 0;
    for (auto& e : ord)
        cntleft += e;
    for (int i = n - 1; i >= 0; i--) {
        if (ord[i] == 0)
            continue;
        int beepnow = 0;
        while (beepnow < i + 1) {
            beepnow += read();
            if (beepnow >= i + 1)
                break;
        }
        wait();
        ord[i]--;
    }
}
```

```
        cntleft--;
        if (cntleft == 0)
            break;
        flip();
        i++;
    }
    end();
}
```

## Task C ()

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <set>
#include <unordered_set>
#include <map>
#include <unordered_map>
#include <random>
#include <assert.h>

using namespace std;

typedef long long int ll;
typedef long double ld;

#define print(a) do { cerr << #a << ":_"; for (auto& e : a) cerr << e << " "; cerr << "\n"; }
while (0);
#define input(a) do { for (auto& e : a) cin >> e; } while (0);

mt19937_64 rnd(1231);
const int MAXINT = 1e9 + 12;
const ll MAXLL = (ll)1e18 + 21;

string swapForward(string s) {
    // 2d
    if (s == "00")
        return "0?";
    if (s == "01")
        return "??";
    if (s == "10")
        return "1?";
    if (s == "11")
        return "?1";
    // 3d
    if (s == "000")
        return "???";
    if (s == "001")
        return "??1";
    if (s == "010")
        return "??0";
    if (s == "011")
        return "?1?";
    if (s == "100")
        return "?0?";
    if (s == "101")
        return "10?";
    if (s == "110")
        return "1?0";
    if (s == "111")
        return "?11";
}

string swapBackward(string s) {
    // 2d
    if (s == "0?")
        return "00";
    if (s == "??")
        return "01";
    if (s == "1?")
        return "10";
    if (s == "?1")
        return "11";
    // 3d
    if (s == "???")
        return "000";
    if (s == "??1")
        return "001";
    if (s == "??0")
        return "010";
    if (s == "?1?")
        return "011";
    if (s == "?0?")
        return "100";
}
```

```

        if (s == "10?")
            return "101";
        if (s == "1?0")
            return "110";
        if (s == "?11")
            return "111";
    }

    void encode(string& s) {
        int i0 = 0;
        if (s.length() % 2 == 1) {
            string tocode = "";
            tocode += s[0];
            tocode += s[1];
            tocode += s[2];
            tocode = swapForward(tocode);
            s[0] = tocode[0];
            s[1] = tocode[1];
            s[2] = tocode[2];
            i0 = 3;
        }
        for (int i = i0; i + 1 < (int)s.length(); i += 2) {
            string tocode = "";
            tocode += s[i];
            tocode += s[i + 1];
            tocode = swapForward(tocode);
            s[i] = tocode[0];
            s[i + 1] = tocode[1];
        }
    }

    void decode(string& s) {
        int i0 = 0;
        if (s.length() % 2 == 1) {
            string tocode = "";
            tocode += s[0];
            tocode += s[1];
            tocode += s[2];
            tocode = swapBackward(tocode);
            s[0] = tocode[0];
            s[1] = tocode[1];
            s[2] = tocode[2];
            i0 = 3;
        }
        for (int i = i0; i + 1 < (int)s.length(); i += 2) {
            string tocode = "";
            tocode += s[i];
            tocode += s[i + 1];
            tocode = swapBackward(tocode);
            s[i] = tocode[0];
            s[i + 1] = tocode[1];
        }
    }

    void solve() {
        string now;
        cin >> now;
        int cntq = 0;
        for (auto& e : now)
            if (e == '?')
                cntq++;

        if (cntq == 0)
            encode(now);
        else
            decode(now);
        cout << now << "\n";
    }

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

        int t;
        cin >> t;
        while (t--) {

```

```
        }  
    }  
    solve();  
}
```

## Task D ()

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <set>
#include <unordered_set>
#include <map>
#include <unordered_map>
#include <random>
#include <assert.h>

using namespace std;

#pragma GCC optimize("Ofast")

typedef long long int ll;
typedef long double ld;

#define print(a) do { cerr << #a << ":\n"; for (auto& e : a) cerr << e << "\n"; cerr << "\n"; } while (0);
#define input(a) do { for (auto& e : a) cin >> e; } while (0);

mt19937_64 rnd(123);
const int MAXINT = 1e9 + 12;
const ll MAXLL = (ll)1e18 + 21;

const ll MOD = 998244353;
//const ll MOD = 37;

#define int ll

ll fastpow(ll x, ll pow) {
    if (pow == 0)
        return 1;
    if (pow == 1)
        return x % MOD;
    if (pow == 2) {
        return ((x % MOD) * x) % MOD;
    }
    if (pow % 2 == 0) {
        return fastpow(fastpow(x, pow / 2), 2);
    }
    return (x * fastpow(x, pow - 1)) % MOD;
}

ll calcfactor(ll x) {
    if (x == 0)
        return 1;
    if (x < MOD) {
        if (x <= MOD / 2) {
            ll ans = 1;
            for (ll i = 2; i <= x; i++) {
                ans = (ans * i) % MOD;
            }
            return ans;
        }
        else {
            ll bads = 1;
            for (ll i = x + 1; i < MOD; i++) {
                bads = (bads * i) % MOD;
            }
            bads = fastpow(bads, MOD - 2);
            ll ans = ((MOD - 1) * bads) % MOD;
            return ans;
        }
    }
    ll ansnow = 1;
    ll cntzero = x / MOD;
    if (cntzero % 211 == 0) {
        ansnow = 1;
    }
    else
        ansnow = MOD - 1;
}
```



```

        //ansnow = (ansnow * calcfactor(x / MOD)) % MOD;
        ansnow = (ansnow * calcfactor(x % MOD)) % MOD;
        return ansnow;
    }

    signed main() {
        ios::sync_with_stdio(0);
        cin.tie(0); cout.tie(0);

        ll n;
        cin >> n;
        n++;

        ll cntpow = n / MOD;
        //assert(cntpow == 0 || cntpow % 2 == 1);

        ll cntzero = n / MOD + (n / MOD / MOD);
        ll ans = calcfactor(n);
        cout << cntzero << " " << ans << "\n";
    }
}

```

## Task E ()

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <set>
#include <unordered_set>
#include <map>
#include <unordered_map>
#include <random>
#include <assert.h>

using namespace std;

typedef long long int ll;
typedef long double ld;

#define print(a) do { cerr << #a << ":_"; for (auto& e : a) cerr << e << " "; cerr << "\n"; } \
while (0);
#define input(a) do { for (auto& e : a) cin >> e; } while (0);

mt19937_64 rnd(1231);
const int MAXINT = 1e9 + 12;
const ll MAXLL = (ll)1e18 + 21;

struct query {
    int a = 0;
    int b = 0;
    int d = 0;
    int id = 0;
};

vector<int> getdels(int d) {
    vector<int> ans;
    for (int i = 1; i * i <= d; i++) {
        if (d % i == 0) {
            ans.push_back(i);
            if (i * i != d)
                ans.push_back(d / i);
        }
    }
    return ans;
}

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

    int n;
    cin >> n;
    vector<int> ord(n);
    input(ord);

    int bs = 3000;
    int cntb = n / bs;
    if (n % bs != 0)
        cntb++;

    vector<ll> anses(cntb, 0);
    vector<map<int, int>> cntPerBlock(cntb);
    for (int i = 0; i < n; i++){
        anses[i / bs]++;
        cntPerBlock[i / bs][ord[i]]++;
    }

    int q;
    cin >> q;
    vector<query> queries(q);
    for (int i = 0; i < q; i++) {
        queries[i].id = i;
        cin >> queries[i].a >> queries[i].b >> queries[i].d;
    }
    vector<ll> queryAnses(q);
    sort(queries.begin(), queries.end(), [](query& a, query& b) {
```

```

        if (a.d == b.d)
            return a.id < b.id;
        return a.d < b.d;
    });

    int inow = 0;
    int dnow = 1;
    while (inow < q) {
        if (queries[inow].d <= dnow) {
            ll ans = 0;
            int d = queries[inow].d;
            int l = queries[inow].a, r = queries[inow].b;
            while (l < r && l % bs != 0) {
                ans += d / ord[l];
                if (d % ord[l] != 0)
                    ans++;
                l++;
            }
            while (r % bs != 0 && r > 1) {
                ans += d / ord[r - 1];
                if (d % ord[r - 1] != 0)
                    ans++;
                r--;
            }
            int blf = l / bs;
            int blt = r / bs;
            for (int b = blf; b < blt; b++) {
                ans += anses[b];
            }
            queryAnses[queries[inow].id] = ans;
            inow++;
        }
        else {
            auto dels = getdels(dnow);
            for (int b = 0; b < cntb; b++) {
                for (auto& e : dels) {
                    anses[b] += cntPerBlock[b][e];
                }
            }
            dnow++;
        }
    }
    for (auto& e : queryAnses) {
        cout << e << '\n';
    }
}

```

## Task F ()

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <set>
#include <unordered_set>
#include <map>
#include <unordered_map>
#include <random>
#include <assert.h>

using namespace std;

typedef long long int ll;
typedef long double ld;

#define print(a) do { cerr << #a << ":_"; for (auto& e : a) cerr << e << " "; cerr << "\n"; }
while (0);
#define input(a) do { for (auto& e : a) cin >> e; } while (0);

mt19937_64 rnd(1231);
const int MAXINT = 1e9 + 12;
const ll MAXLL = (ll)1e18 + 21;

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

    int n;
    cin >> n;
    vector<pair<int, int>> lines(n);
    for (auto& e : lines) {
        cin >> e.first >> e.second;
    }

    vector<int> anses(n + 1, 1);
    for (int i = n - 1; i >= 0; i--) {
        ll l = 0;
        ll r = 0;
        for (int j = i; j < n; j++) {
            l += lines[j].first;
            r += lines[j].second;
            if (l <= 0 && r >= 0)
                anses[i] = max(anses[i], anses[j + 1] + 1);
        }
    }
    int ans = 0;
    for (auto& e : anses) {
        ans = max(ans, e);
    }
    cout << ans << "\n";
}
```