

Олимпиада СПбГУ по информатике 2021/22 учебного года

A	B	C	D	E	F	Sum
100	100	100	60	100	31	491

Task A ()

```
/*
A: 66
B: 0
C: 100
D: 60
E: 58          3d mo??
F: 31
*/

#include <iostream>
#include <vector>
#include <algorithm>
#include <map>
#include <string>
#include <cmath>
#include <queue>
#include <unordered_set>
#pragma GCC optimize("unroll-loops")
#pragma GCC optimize("Ofast")
#pragma GCC optimize("no-stack-protector")
#pragma GCC target("sse,sse2,sse3,ssse3,popcnt,abm,mmx,tune=native")
#pragma GCC optimize("fast-math")
#pragma GCC optimize("-O3")
#define int long long
using namespace std;

const int mod = 998244353;

signed main() {
    ios::sync_with_stdio(0);
    cin.tie(0);
    cout.tie(0);
    //freopen("in.txt", "r", stdin);
    //freopen("out.txt", "w", stdout);

    int n; cin >> n;
    vector <pair <int, int>> a(n);
    map <int, int> q;
    int mnz = 1e9 + 1;
    for (int i = 0; i < n; i++) {
        int v, z; cin >> v >> z;
        while (v % 10 == 0) {
            v /= 10;
            z++;
        }
        a[i] = { z, v };
        mnz = min(z, mnz);
        q[z] += v;
    }

    for (int j = mnz; ; j++) {
        int z = 0;
        int v = q[j];
        if (v == 0)
```

```
        continue;
    while (v % 10 == 0) {
        v /= 10;
        z++;
    }
    if (z == 0) {
        cout << j;
        return 0;
    }
    q[j + z] += v;
}

return 0;
}
```

Task B ()

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <string>
using namespace std;

signed main() {
    int n; cin >> n;
    vector<int> k(n);
    for (int i = 0; i < n; i++) {
        cin >> k[i];
    }

    int j = n - 1;
    while (true) {
        while (j >= 0 && k[j] == 0)
            j--;
        if (j < 0) {
            cout << "Stop" << endl;
            return 0;
        }
        cout << "Flip_and_wait" << endl;
        int cnt = 0;
        while (true) {
            string s; cin >> s;
            cnt += (s.size() - 3) / 2;
            if (cnt > j) {
                k[j]--;
                break;
            }
            cout << "Wait" << endl;
        }
    }

    return 0;
}
```

Task C ()

```
/*
21inf4481
himusheme
```

```
A: 66
B: 0
C: 0
D: 0
E: 0
F: 0
```

```
*/

#include <iostream>
#include <vector>
#include <algorithm>
#include <map>
#include <string>
#include <cmath>
#include <queue>
#include <unordered_set>
#pragma GCC optimize("unroll-loops")
#pragma GCC optimize("Ofast")
#pragma GCC optimize("no-stack-protector")
#pragma GCC target("sse,sse2,sse3,ssse3,popcnt,abm,mmx,tune=native")
#pragma GCC optimize("fast-math")
#pragma GCC optimize("-O3")
// #define int long long
using namespace std;

void solve() {
    string s; cin >> s;
    int n = s.size();

    if (s[0] != '?' && s[1] != '?') {
        string t = s;
        for (int i = 0; i + 1 < n; i += 2) {
            string k = string(1, s[i]) + string(1, s[i + 1]);
            if (k == "00") {
                t[i + 1] = '?';
            }
            else if (k == "01") {
                t[i] = '?';
            }
            else if (k == "10") {
                t[i] = '?';
            }
            else {
                t[i] = '?';
                t[i + 1] = '?';
            }
        }
        cout << t << '\n';
        return;
    }

    string t = s;
    for (int i = 0; i + 1 < n; i += 2) {
        string k = string(1, s[i]) + string(1, s[i + 1]);
        if (k == "0?") {
            t[i + 1] = '0';
        }
        else if (k == "?1") {
            t[i] = '0';
        }
        else if (k == "?0") {
            t[i] = '1';
        }
        else {
            t[i] = '1';
            t[i + 1] = '1';
        }
    }
}
```

```

    }
}
cout << t << '\n';
return;
}

signed main() {
    ios::sync_with_stdio(0);
    cin.tie(0);
    cout.tie(0);
    //freopen("in.txt", "r", stdin);
    //freopen("out.txt", "w", stdout);

    int t = 1;
    cin >> t;
    while (t--) {
        solve();
    }

    return 0;
}

```

Task D ()

```
/*
A: 66
B: 0
C: 100
D: 40
E: 58          3d mo??
F: 31


---



*/

#include <iostream>
#include <vector>
#include <algorithm>
#include <map>
#include <string>
#include <cmath>
#include <queue>
#include <unordered_set>
#pragma GCC optimize("unroll-loops")
#pragma GCC optimize("Ofast")
#pragma GCC optimize("no-stack-protector")
#pragma GCC target("sse,sse2,sse3,ssse3,popcnt,abm,mmx,tune=native")
#pragma GCC optimize("fast-math")
#pragma GCC optimize("-O3")
using namespace std;

const int mod = 998244353;

signed main() {
    ios::sync_with_stdio(0);
    cin.tie(0);
    cout.tie(0);
    //freopen("in.txt", "r", stdin);
    //freopen("out.txt", "w", stdout);

    int n; cin >> n;

    vector<int> dp(n + 1);
    long long a = 1;
    long long b = 2;
    long long c;
    for (int i = 2; i <= n; i++) {
        c = (a * i % mod * i % mod + b) % mod;
        a = b;
        b = c;
    }
    cout << "0_" << b;

    return 0;
}
```

Task E ()

```
/*
A: 100
B: 0 (100?)
C: 100
D: 60
E: 58 (100?)
F: 31


---


*/

#include <iostream>
#include <vector>
#include <algorithm>
// #pragma GCC optimize("unroll-loops")
// #pragma GCC optimize("Ofast")
// #pragma GCC optimize("no-stack-protector")
// #pragma GCC target("sse,sse2,sse3,ssse3,popcnt,abm,mmx,tune=native")
// #pragma GCC optimize("fast-math")
// #pragma GCC optimize("-O3")
#define int long long
using namespace std;

const int mod = 998244353;
const int mx = 1e6 + 1;
const int k = 1000;

vector<int> f(mx);
vector<int> cnt(k);

int get(int i) {
    int res = 0;
    for (int j = 1; j < k; j++) {
        if (j > i)
            break;
        res += cnt[j] * (i / j);
    }
    for (int j = i; j >= 0; j = (j & (j + 1)) - 1) {
        res += f[j];
    }
    return res;
}

void add(int i) {
    for (int j = i; j < mx; j = (j | (j + 1))) {
        f[j]++;
    }
}

void add_element(int x) {
    if (x < k) {
        cnt[x]++;
        return;
    }
    for (int j = x; j < mx; j += x) {
        add(j);
    }
}

signed main() {
    ios::sync_with_stdio(0);
    cin.tie(0);
    cout.tie(0);
    // freopen("in.txt", "r", stdin);
    // freopen("out.txt", "w", stdout);

    int n; cin >> n;
    vector<int> a(n);
    for (int i = 0; i < n; i++) {
        cin >> a[i];
    }
}
```

```

int q; cin >> q;
vector <pair <int, int>> rem(q);
vector <int> pl(q), pr(q);
vector <vector <pair<int, pair <int, int>>>> zapr(n); // {d, 0/1} 0 - left, 1 - right
for (int i = 0; i < q; i++) {
    int l, r, x; cin >> l >> r >> x; x--;
    rem[i] = { l, r };
    l--, r--;
    if (l >= 0)
        zapr[l].push_back({ i, { x, 0 } });
    if (r >= 0)
        zapr[r].push_back({ i, { x, 1 } });
}
for (int i = 0; i < n; i++) {
    add_element(a[i]);
    for (auto& e : zapr[i]) {
        int ind = e.first;
        int d = e.second.first;
        int lr = e.second.second;

        int res = get(d);
        //int res = 0;
        //for (int j = 0; j <= i; j++) {
        //    res += d / a[j];
        //}

        if (lr == 0)
            pl[ind] = res;
        else
            pr[ind] = res;
    }
}
for (int i = 0; i < q; i++) {
    int l = rem[i].first, r = rem[i].second;
    cout << (r - l) + (pr[i] - pl[i]) << '\n';
}

return 0;
}

```


Task F ()

```
/*
A: 66
B: 0
C: 100
D: 0
E: 58
F: 0
-----
224
*/

#include <iostream>
#include <vector>
#include <algorithm>
#include <map>
#include <string>
#include <cmath>
#include <queue>
#include <unordered_set>
#pragma GCC optimize("unroll-loops")
#pragma GCC optimize("Ofast")
#pragma GCC optimize("no-stack-protector")
#pragma GCC target("sse,sse2,sse3,ssse3,popcnt,abm,mmx,tune=native")
#pragma GCC optimize("fast-math")
#pragma GCC optimize "-O3"
#define int long long
using namespace std;

const int k = 1000;

signed main() {
    ios::sync_with_stdio(0);
    cin.tie(0);
    cout.tie(0);
    //freopen("in.txt", "r", stdin);
    //freopen("out.txt", "w", stdout);

    int n; cin >> n;
    vector <pair <int, int>> a(n);
    for (int i = 0; i < n; i++) {
        cin >> a[i].first >> a[i].second;
    }
    vector <int> dp(n + 1, 1); dp[0] = 1;
    for (int i = 1; i <= n; i++) {
        int l = 0, r = 0;
        for (int j = i - 1; j >= 0; j--) {
            l += a[j].first;
            r += a[j].second;
            if (l <= 0 && 0 <= r)
                dp[i] = max(dp[i], dp[j] + 1);
        }
    }

    int ans = 1;
    for (int i = 1; i <= n; i++) {
        ans = max(ans, dp[i]);
    }
    cout << ans;

    return 0;
}
```