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 <bits/stdc++.h>

using namespace std;

const int Inf = 2e9;

int main()
{
    int n;
    cin >> n;
    map<int, int> dict;
    for (int i = 0; i < n; ++i) {
        string a;
        int b;
        cin >> a;
        cin >> b;
        reverse(a.begin(), a.end());
        for (int i = 0; i < a.size(); ++i) {
            if (dict.find(i + b) == dict.end()) {
                dict[i + b] = a[i] - '0';
            }
            else {
                dict[i + b] += a[i] - '0';
                int l = i + b;
                while (dict.find(l + 1) != dict.end() && dict[l] >= 10) {
                    dict[l] -= 10;
                    l++;
                    dict[l]++;
                }
                if (dict.find(l + 1) == dict.end() && dict[l] >= 10) {
                    dict[l + 1] = 1;
                    dict[l] -= 10;
                }
            }
        }
    }
    int minim = Inf;
    for (auto i: dict) {
        if (i.second != 0) {
            minim = min(minim, i.first);
        }
    }
    cout << minim;
    return 0;
}
```

Task B ()

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

using namespace std;

const int Inf = 2e9;

int get_numb(string s) {
    return (s.size() - 3) / 2;
}

int main()
{
    int n;
    cin >> n;
    int data[n];
    int last_ind_data = 0;
    int summa = 0;
    for (int i = 0; i < n; ++i) {
        cin >> data[i];
        summa += data[i];
        if (data[i] != 0) {
            last_ind_data = i;
        }
    }
    vector<pair<int, int> > otr;
    int a = -1;
    int b = -1;
    cout << "Flip␣and␣wait" << endl;
    string s;
    while (b < last_ind_data) {
        cin >> s;
        a = b + 1;
        b = a + get_numb(s) - 1;
        otr.push_back({a, b});
        if (b < last_ind_data) {
            cout << "Wait" << endl;
        }
    }
    data[last_ind_data]--;
    summa--;
    if (summa == 0) {
        cout << "Stop" << endl;
        return 0;
    }
    cout << "Flip␣and␣wait" << endl;
    int l = otr.size() - 1;
    for (int i = last_ind_data; i >= 0; --i) {
        if (i < otr[l].first) {
            l--;
        }
        for (int j = 0; j < data[i]; ++j) {
            for (int k = 0; k < l; ++k) {
                cin >> s;
                cout << "Wait" << endl;
            }
            summa--;
            cin >> s;
            if (summa == 0) {
                cout << "Stop" << endl;
                return 0;
            }
            cout << "Flip␣and␣wait" << endl;
        }
    }
    cout << "Stop" << endl;
    return 0;
}
```

Task C ()

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

using namespace std;

const int Inf = 2e9;

string code_ne(string s) {
    int one = 0;
    int two = 0;
    for (int i = 0; i < s.size(); ++i) {
        if (s[i] == '1') {
            one++;
        }
        else {
            two++;
        }
    }
    if (one == 0 || two == 0) {
        string s1;
        for (int i = 0; i < s.size() / 2; ++i) {
            s1 += '?';
        }
        for (int i = s.size() / 2; i < s.size(); ++i) {
            s1 += s[i];
        }
        return s1;
    }
    string s1;
    for (int i = 0; i < s.size(); ++i) {
        if (one < two && s[i] == '0') {
            s1 += '?';
        }
        else if (one > two && s[i] == '1') {
            s1 += '?';
        }
        else {
            s1 += s[i];
        }
    }
    return s1;
}

string code_da(string s) {
    if (s.size() == 2) {
        if (s == "11") {
            return "1?";
        }
        if (s == "10") {
            return "?0";
        }
        if (s == "01") {
            return "?1";
        }
        if (s == "00") {
            return "0?";
        }
    }
    int one = 0;
    int two = 0;
    for (int i = 0; i < s.size(); ++i) {
        if (s[i] == '1') {
            one++;
        }
        else {
            two++;
        }
    }
    if (one == 0 || two == 0) {
        string s1;
        for (int i = 0; i < s.size() / 2; ++i) {
            s1 += '?';
        }
        for (int i = s.size() / 2; i < s.size(); ++i) {
            s1 += s[i];
        }
        return s1;
    }
    if (one == two) {
        string s1;
        for (int i = 0; i < s.size(); ++i) {
            if (s[i] != s[0]) {
                s1 += '?';
            }
            else {
                s1 += s[i];
            }
        }
        return s1;
    }
    string s1;
    for (int i = 0; i < s.size(); ++i) {
        if (one < two && s[i] == '0') {
            s1 += '?';
        }
        else if (one > two && s[i] == '1') {
            s1 += '?';
        }
        else {
            s1 += s[i];
        }
    }
    return s1;
}

string decode_ne(string s) {
    int c;
    int q = 0;
    for (int i = 0; i < s.size(); ++i) {
        if (s[i] == '?') {
            q++;
        }
        else {
            c = s[i] - '0';
        }
    }
    if (q * 2 < s.size()) {
        string s1;
        for (int i = 0; i < s.size(); ++i) {
            s1 += c + '0';
        }
        return s1;
    }
    string s1;
    for (int i = 0; i < s.size(); ++i) {
        if (s[i] == '?') {
            s1 += (1 - c) + '0';
        }
        else {
            s1 += s[i];
        }
    }
    return s1;
}

string decode_da(string s) {
    if (s.size() == 2) {
        if (s == "1?") {
            return "11";
        }
        if (s == "0?") {
            return "00";
        }
        if (s == "?0") {
            return "10";
        }
        if (s == "?1") {
            return "01";
        }
    }
    int one = 0;
    int two = 0;
    int q = 0;
    for (int i = 0; i < s.size(); ++i) {
        if (s[i] == '0') {
            two++;
        }
        else if (s[i] == '1') {
            one++;
        }
        else {
            q++;
        }
    }
    if ((one == 0 || two == 0) && s[0] == '?' && (q == one + two)) {
        string s1;
        for (int i = 0; i < s.size(); ++i) {
            if (s[i] == '?') {
                if (two == 0) {
                    s1 += '1';
                }
                else {
                    s1 += '0';
                }
            }
            else {
                s1 += s[i];
            }
        }
        return s1;
    }
    if ((one == 0 || two == 0) && (q == one + two)) {
        string s1;
        for (int i = 0; i < s.size(); ++i) {
            if (s[i] == '?') {
                if (two == 0) {
                    s1 += '0';
                }
                else {
                    s1 += '1';
                }
            }
            else {
                s1 += s[i];
            }
        }
        return s1;
    }
    string s1;
    for (int i = 0; i < s.size(); ++i) {
        if (s[i] == '?') {
            if (one == 0) {
                s1 += '1';
            }
            else {
                s1 += '0';
            }
        }
        else {
            s1 += s[i];
        }
    }
    return s1;
}

string decode(string s) {
    if (s.size() % 2 == 1) {
        return decode_ne(s);
    }
    else {
        return decode_da(s);
    }
}

string code(string s) {
    if (s.size() % 2 == 1) {
        return code_ne(s);
    }
    else {
        return code_da(s);
    }
}

int main()
{
    int n;
    cin >> n;
    for (int i = 0; i < n; ++i) {
        string s;
        cin >> s;
        bool fl = true;
        for (int j = 0; j < s.size(); ++j) {
            if (s[j] == '?') {
                fl = false;
                break;
            }
        }
        if (fl) {
            cout << code(s) << endl;
        }
        else {
            cout << decode(s) << endl;
        }
    }
    return 0;
}
```

Task D ()

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

using namespace std;

const int Inf = 2e9;
const long long mod = 998244353;

int ddata[10] = {808258749, 185776952, 832058496, 859017922, 725936912, 475217179, 143573352,
    943785935, 378507008};

long long fact(long long n) {
    long long cnt = 1;
    int p = 0;
    long long k = 100000000;
    while (k <= n) {
        cnt *= ddata[p];
        cnt %= mod;
        p++;
        k += 100000000;
    }
    k -= 100000000;
    for (long long i = k + 1; i <= n; ++i) {
        cnt *= i;
        cnt %= mod;
    }
    return cnt;
}

long long do_(long long l, long long r) {
    long long cnt =1;
    for (long long i = l; i <= r; ++i) {
        cnt *=i;
        cnt %= mod;
    }
    return cnt;
}

int main()
{
    //cout << do_(9 * 100000000 + 1, mod - 1);
    long long n;
    cin >> n;
    n++;
    long long cnt1 = 0;
    long long k = mod;
    while (n >= k) {
        cnt1 += n / k;
        k *= mod;
    }
    long long ost = 1;
    if (n / mod % 2 == 1) {
        ost = mod - 1;
    }
    n %= mod;
    ost *= fact(n);
    ost %= mod;
    cout << cnt1 << '␣' << ost;
    return 0;
}
```

Task E ()

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

using namespace std;

int main()
{
    int n, q;
    cin >> n;
    int data[n];
    for (int i = 0; i < n; ++i) {
        cin >> data[i];
    }
    cin >> q;
    for (int i = 0; i < q; ++i) {
        int a, b, d;
        cin >> a;
        cin >> b;
        cin >> d;
        int ans = 0;
        for (int j = a; j < b; ++j) {
            ans += (d + data[j] - 1) / data[j];
        }
        cout << ans << endl;
    }
    return 0;
}
```

Task F ()

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

using namespace std;

const int Inf = 2e9;
const long long mod = 998244353;

int main()
{
    long long n;
    cin >> n;
    pair<long long, long long> data[n];
    for (long long i = 0; i < n; ++i) {
        long long a, b;
        cin >> a;
        cin >> b;
        data[i] = {a, b};
    }
    long long dp[n + 1];
    dp[0] = 1;
    for (long long i = 1; i <= n; ++i) {
        dp[i] = 1;
        long long minim = 0;
        long long maxim = 0;
        for (long long j = i - 1; j >= 0; --j) {
            minim += data[j].first;
            maxim += data[j].second;
            if (minim <= 0 && 0 <= maxim) {
                dp[i] = max(dp[i], dp[j] + 1);
            }
        }
    }
    long long ans = 0;
    for (long long i = 0; i <= n; ++i) {
        ans = max(ans, dp[i]);
    }
    cout << ans;
    return 0;
}
```