Олимпиада СПбГУ по информатике 2021/22 учебного года

A	B	C	D	E	F	Sum
100	100	100	60	58	31	449

Task A ()

```
#include <iostream>
#include <algorithm>

const int maxN = 2e5;

struct Item {
        int a, b;
        friend std::istream& operator>>(std::istream& is, Item& a) {
                is >> a.a >> a.b;
                while (a.a % 10 == 0) {
                        a.a /= 10;
                        a.b++;
                }
                return is;
        }
        bool operator<(const Item& other) const {
                return b < other.b || (b == other.b && a < other.a);
        }
} a[maxN];

bool f = true;
int n, ans, it;
long long temp;

int main() {
        std::ios_base::sync_with_stdio(false);
        std::cin.tie(0);
        std::cout.tie(0);
        std::cin >> n;
        for (int i = 0; i < n; i++)
                std::cin >> a[i];
        std::sort(a, a + n);
        for (int i = 0; i < n && f; i++) {
                if (it < a[i].b) {
                        while (it < a[i].b && temp > 0 && f) {
                                if (temp % 10 == 0)
                                        ans++;
                                else
                                        f = false;
                                temp /= 10;
                                it++;
                        }
                        if (!f)
                                break;
                        if (it < a[i].b) {
                                ans += a[i].b - it;
                                it = a[i].b;
                        }
                }
                temp += a[i].a;
        }
        while (f && temp % 10 == 0) {
                temp /= 10;
                ans++;
        }
        std::cout << ans;
        return 0;
}
```

Task B ()

```
#include <iostream>
#include <string>

const int maxN = 100;

int n, k[maxN], cnt;
std::string s;

int main() {
        std::cin >> n;
        for (int i = 0; i < n; i++)
                std::cin >> k[i];
        for (int i = n - 1; i >= 0; i--)
                for (int j = 0; j < k[i]; j++) {
                        std::cout << "Flip␣and␣wait" << std::endl;
                        cnt = 0;
                        std::cin >> s;
                        cnt += (s.size() - 3) / 2;
                        while (cnt <= i) {
                                std::cout << "Wait" << std::endl;
                                std::cin >> s;
                                cnt += (s.size() - 3) / 2;
                        }
                }
        std::cout << "Stop" << std::endl;
        return 0;
}
```

Task C ()

```
#include <iostream>
#include <string>

bool f;
int t, cnt, ones, zeros;
std::string s;

int main() {
        std::ios_base::sync_with_stdio(false);
        std::cin.tie(0);
        std::cout.tie(0);
        std::cin >> t;
        while (t--) {
                std::cin >> s;
                for (int i = 0; i < s.size() && !f; i++)
                        f = (s[i] == '?');
                if (f) {
                        cnt = 0;
                        for (int i = 0; i < s.size(); i++)
                                cnt += (s[i] == '?');
                        for (int i = 0; i < s.size(); i++)
                                if (s[i] == '?')
                                        std::cout << (cnt & 1);
                                else
                                        std::cout << s[i];
                        std::cout << "\n";
                }
                else {
                        ones = zeros = 0;
                        for (int i = 0; i < s.size(); i++) {
                                ones += (s[i] == '1');
                                zeros += (s[i] == '0');
                        }
                        if (ones == zeros) {
                                if (ones & 1)
                                        zeros--;
                                else
                                        ones--;
                        }
                        if (ones > zeros) {
                                ones -= 1 - (ones & 1);
                                for (int i = 0; i < s.size(); i++)
                                        if (s[i] == '1' && ones) {
                                                std::cout << '?';
                                                ones--;
                                        }
                                        else
                                                std::cout << s[i];
                        }
                        else if (ones < zeros) {
                                zeros -= (zeros & 1);
                                for (int i = 0; i < s.size(); i++)
                                        if (s[i] == '0' && zeros) {
                                                std::cout << '?';
                                                zeros--;
                                        }
                                        else
                                                std::cout << s[i];
                        }
                        std::cout << "\n";
                }
        }
        return 0;
}
```

Task D ()

```
#include <iostream>

const int mod = 998244353;

int n;
long long ans = 1;

int main() {
        std::cin >> n;
        for (int i = 1; i <= n + 1; i++)
                ans = ans * i % mod;
        std::cout << 0 << "␣" << ans << "\n";
        return 0;
}
```

Task E ()

```
#include <iostream>
#include <cmath>
#include <algorithm>

const long long maxN = 3e5 + 100, SZ = 1400;

long long K, it;

struct Query {
        long long a, b, d, id;
        friend std::istream& operator>>(std::istream& is, Query& q) {
                is >> q.a >> q.b >> q.d;
                q.id = it++;
                return is;
        }
        bool operator<(const Query& other) const {
                long long b1 = a / K, b2 = other.a / K;
                if (b1 == b2) {
                        if (b1 & 1)
                                return b > other.b;
                        else
                                return b < other.b;
                }
                return b1 < b2;
        }
} a[maxN];

long long n, u[maxN], q, b[SZ], t[maxN + 1], l, r;
long long ans[maxN];

void insert(long long i, long long x) {
        for (; i <= maxN; i |= (i + 1))
                t[i] += x;
}

long long get(long long i) {
        long long res = 0;
        for (; i >= 0; i = (i & (i + 1)) - 1)
                res += t[i];
        return res;
}

int main() {
        std::ios_base::sync_with_stdio(false);
        std::cin.tie(0);
        std::cout.tie(0);
        std::cin >> n;
        for (long long i = 0; i < n; i++)
                std::cin >> u[i];
        std::cin >> q;
        for (long long i = 0; i < q; i++)
                std::cin >> a[i];
        K = std::max((long long)1, (long long)(n / std::sqrt(q)));
        std::sort(a, a + q);
        l = r = a[0].a;
        for (long long i = 0; i < q; i++) {
                while (l > a[i].a) {
                        l--;
                        if (u[l] < SZ)
                                b[u[l]]++;
                        else
                                insert(u[l], 1);
                }
                while (r < a[i].b) {
                        if (u[r] < SZ)
                                b[u[r]]++;
                        else
                                insert(u[r], 1);
                        r++;
                }
                while (l < a[i].a) {
                        if (u[l] < SZ)
```

```
				b[u[l]]--;
			else
				insert(u[l], -1);
			l++;
		}
		while (r > a[i].b) {
			r--;
			if (u[r] < SZ)
				b[u[r]]--;
			else
				insert(u[r], -1);
		}
		for (long long cur = maxN, next = 0, temp = 0; cur >= SZ; cur = next - 1) {
			temp = (a[i].d + cur - 1) / cur;
			next = (a[i].d + temp - 1) / temp;
			ans[a[i].id] += 1LL * (get(cur) - get(next - 1)) * temp;
		}
		for (long long cur = SZ - 1; cur > 0; cur--)
			ans[a[i].id] += 1LL * ((a[i].d + cur - 1) / cur) * b[cur];
	}
	for (long long i = 0; i < q; i++)
		std::cout << ans[i] << "\n";
	return 0;
}
```

Task F ()

```
#include <iostream>

const int maxN = 1e6;

int n, l[maxN], r[maxN], dp[maxN + 1], ans;
long long l0, r0;

int main() {
        std::ios_base::sync_with_stdio(false);
        std::cin.tie(0);
        std::cout.tie(0);
        std::cin >> n;
        for (int i = 0; i < n; i++)
                std::cin >> l[i] >> r[i];
        for (int i = n; i >= 0; i--) {
                dp[i] = 1;
                l0 = r0 = 0;
                for (int j = i; j < n; j++) {
                        l0 += l[j];
                        r0 += r[j];
                        if (l0 <= 0 && 0 <= r0)
                                dp[i] = std::max(dp[i], dp[j + 1] + 1);
                }
                ans = std::max(ans, dp[i]);
        }
        std::cout << ans;
        return 0;
}
```