Олимпиада СПбГУ по информатике 2021/22 учебного года

A	B	C	D	E	F	Sum
100	100	100	80	58	98	536

Task A ()

```
#include <algorithm>
#include <cmath>
#include <cstring>
#include <iomanip>
#include <iostream>
#include <map>
#include <numeric>
#include <queue>
#include <set>
#include <string>
#include <unordered_map>
#include <unordered_set>
#include <vector>
#include <ctime>
#include <cassert>

#define all(a) (a).begin(), (a).end()
#define ssize(a) (int)(a).size()

using namespace std;

using ll = long long;


int main() {
#ifdef LOCAL
   freopen("input.txt", "r", stdin);
#endif
  ios_base::sync_with_stdio(false);
  cin.tie(nullptr);

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

  map<int, ll> sum;
  for (int i = 0; i < n; ++i) {
    sum[b[i]] += a[i];
  }

  for (auto &[i, j] : sum) {
    if (j % 10 != 0) {
      cout << i;
      break;
    }
    sum[i + 1] += j / 10;
  }


  return 0;
}
```

Task B ()

```
#include <algorithm>
#include <cassert>
#include <cmath>
#include <cstring>
#include <ctime>
#include <iomanip>
#include <iostream>
#include <map>
#include <numeric>
#include <queue>
#include <set>
#include <string>
#include <unordered_map>
#include <unordered_set>
#include <vector>

#define all(a) (a).begin(), (a).end()
#define ssize(a) (int)(a).size()

using namespace std;

using ll = long long;

int main() {
#ifdef LOCAL
  //freopen("input.txt", "r", stdin);
#endif
  ios_base::sync_with_stdio(false);
  cin.tie(nullptr);

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

  while (true) {
    int mx = n - 1;
    while (mx >= 0 && k[mx] == 0) {
      --mx;
    }
    if (mx == -1) {
      break;
    }

    cout << "Flip␣and␣wait" << endl;

    int cnt = 0;
    while (true) {
      string info;
      cin >> info;
      assert(info[0] == 'B');
      cnt += count(all(info), 'e') / 2;
      if (cnt >= mx + 1) {
        break;
      } else {
        cout << "Wait" << endl;
      }
    }

    --k[mx];
  }

  cout << "Stop" << endl;

  return 0;
}
```

Task C ()

```
#include <algorithm>
#include <cassert>
#include <cmath>
#include <cstring>
#include <ctime>
#include <iomanip>
#include <iostream>
#include <map>
#include <numeric>
#include <queue>
#include <random>
#include <set>
#include <string>
#include <unordered_map>
#include <unordered_set>
#include <vector>

#define all(a) (a).begin(), (a).end()
#define ssize(a) (int)(a).size()

using namespace std;

using ll = long long;


map<string, string> f;
map<string, string> g;


void solve() {
  string s;
  cin >> s;

  int n = ssize(s);

  int cntq = count(all(s), '?');

  if (cntq == 0) {
    for (int i = 0; i + 1 < n; i += 2) {
      string tmp = f[s.substr(i, 2)];
      s[i] = tmp[0];
      s[i + 1] = tmp[1];
    }
  } else {
    for (int i = 0; i + 1 < n; i += 2) {
      string tmp = g[s.substr(i, 2)];
      s[i] = tmp[0];
      s[i + 1] = tmp[1];
    }
  }
  cout << s << "\n";
}

int main() {
#ifdef LOCAL
  freopen("input.txt", "r", stdin);
#endif
  ios_base::sync_with_stdio(false);
  cin.tie(nullptr);

  f = {{"00", "0?"}, {"01", "?1"}, {"10", "?0"}, {"11", "1?"}};
  for (auto &[x, y] : f) {
    g[y] = x;
  }

  int t;
  cin >> t;
  while (t--) {
    solve();
  }
  return 0;
}
```

Task D ()

```cpp
#include <algorithm>
#include <cassert>
#include <cmath>
#include <cstring>
#include <ctime>
#include <iomanip>
#include <iostream>
#include <map>
#include <numeric>
#include <queue>
#include <set>
#include <string>
#include <unordered_map>
#include <unordered_set>
#include <vector>

#define all(a) (a).begin(), (a).end()
#define ssize(a) (int)(a).size()

using namespace std;

using ll = long long;

const ll mod = 998244353;

map<int, int> mp{{0, 1},
    {5000000, 428937595},
    {10000000, 295201906},
    {15000000, 813259672},
    {20000000, 160030060},
    {25000000, 433880730},
    {30000000, 957629942},
    {35000000, 31729117},
    {40000000, 545208507},
    {45000000, 767818091},
    {50000000, 213689172},
    {55000000, 363266670},
    {60000000, 760025067},
    {65000000, 117371703},
    {70000000, 939830261},
    {75000000, 770845925},
    {80000000, 506268060},
    {85000000, 328979964},
    {90000000, 39806322},
    {95000000, 512554988},
    {100000000, 808258749},
    {105000000, 694814243},
    {110000000, 440133909},
    {115000000, 385510728},
    {120000000, 686156489},
    {125000000, 222059821},
    {130000000, 741797144},
    {135000000, 677432735},
    {140000000, 390377694},
    {145000000, 339045014},
    {150000000, 12629586},
    {155000000, 634938288},
    {160000000, 544711799},
    {165000000, 377929821},
    {170000000, 104121967},
    {175000000, 313632531},
    {180000000, 495867250},
    {185000000, 865854329},
    {190000000, 421290700},
    {195000000, 386147469},
    {200000000, 117153405},
    {205000000, 18536028},
    {210000000, 57084755},
    {215000000, 325341339},
    {220000000, 202713771},
    {225000000, 546969497},
    {230000000, 675932866},
{235000000, 759880589},
{240000000, 79781699},
{245000000, 848259019},
{250000000, 956276337},
{255000000, 941778993},
{260000000, 652678397},
{265000000, 315094878},
{270000000, 35212756},
{275000000, 433186147},
{280000000, 655645460},
{285000000, 64748124},
{290000000, 468129309},
{295000000, 281295633},
{300000000, 761699708},
{305000000, 600130702},
{310000000, 533047427},
{315000000, 676886948},
{320000000, 287671032},
{325000000, 423996017},
{330000000, 206068022},
{335000000, 882517476},
{340000000, 50865043},
{345000000, 993580422},
{350000000, 144980423},
{355000000, 843705280},
{360000000, 111276893},
{365000000, 26368504},
{370000000, 259415897},
{375000000, 918048438},
{380000000, 444094191},
{385000000, 723617861},
{390000000, 593907889},
{395000000, 173350007},
{400000000, 573994984},
{405000000, 401920943},
{410000000, 892454686},
{415000000, 84924870},
{420000000, 566073550},
{425000000, 874007723},
{430000000, 128761001},
{435000000, 149123648},
{440000000, 888483202},
{445000000, 855479832},
{450000000, 251718753},
{455000000, 659478190},
{460000000, 548033568},
{465000000, 93829695},
{470000000, 428105027},
{475000000, 616996159},
{480000000, 742756734},
{485000000, 633125919},
{490000000, 546182474},
{495000000, 249619035},
{500000000, 62402409},
{505000000, 514588945},
{510000000, 102052166},
{515000000, 962390239},
{520000000, 826426395},
{525000000, 687413213},
{530000000, 159186619},
{535000000, 820097297},
{540000000, 926316039},
{545000000, 850407126},
{550000000, 176055335},
{555000000, 888573766},
{560000000, 51568171},
{565000000, 341370469},
{570000000, 414163604},
{575000000, 600191572},
{580000000, 604947226},
{585000000, 632441623},
{590000000, 681666415},
{595000000, 978860867},
{600000000, 511621808},
{605000000, 631078482},
{610000000, 924112080},
{615000000, 805963889},
{620000000, 265769800},
{625000000, 172922144},
{630000000, 955559118},
{635000000, 576699812},
{640000000, 763148293},
{645000000, 253151120},
{650000000, 472709375},
{655000000, 184639537},
{660000000, 19536133},
{665000000, 687419227},
{670000000, 860830935},
{675000000, 202133852},
{680000000, 290471030},
{685000000, 422332597},
{690000000, 851685235},
{695000000, 84257200},
{700000000, 242726978},
{705000000, 744816299},
{710000000, 169855231},
{715000000, 714502830},
{720000000, 612759169},
{725000000, 158013406},
{730000000, 599797734},
{735000000, 10958310},
{740000000, 961628039},
{745000000, 261743865},
{750000000, 953297493},
{755000000, 539880795},
{760000000, 62806842},
{765000000, 681051568},
{770000000, 37844313},
{775000000, 682765157},
{780000000, 909741023},
{785000000, 551621442},
{790000000, 689361523},
{795000000, 349960501},
{800000000, 887890124},
{805000000, 568925901},
{810000000, 380694152},
{815000000, 109618190},
{820000000, 669317759},
{825000000, 918239352},
{830000000, 367270918},
{835000000, 899305835},
{840000000, 806951470},
{845000000, 664034750},
{850000000, 843736533},
{855000000, 776142608},
{860000000, 377403437},
{865000000, 354033051},
{870000000, 945260111},
{875000000, 239480646},
{880000000, 786127243},
{885000000, 12198440},
{890000000, 80918046},
{895000000, 49743354},
{900000000, 875880304},
{905000000, 636257427},
{910000000, 364983542},
{915000000, 994704486},
{920000000, 623250998},
{925000000, 68064804},
{930000000, 598764068},
{935000000, 278395191},
{940000000, 804930040},
{945000000, 756464212},
{950000000, 24257676},
{955000000, 971716517},
{960000000, 214821357},
{965000000, 707669156},
{970000000, 791011898},
    {975000000, 291032164},
    {980000000, 954947696},
    {985000000, 289232396},
    {990000000, 183092975},
    {995000000, 781587414},
};

ll fact(ll n) {
  auto it = --mp.upper_bound(n);
  ll ans = it->second;
  for (int i = it->first + 1; i <= n; ++i) {
    ans = ans * i % mod;
  }
  return ans;
}

ll binpow(ll a, ll p) {
  if (p == 0) {
    return 1;
  }
  if (p % 2 == 0) {
    return binpow(a * a % mod, p / 2);
  }
  return a * binpow(a, p - 1) % mod;
}


int main() {
#ifdef LOCAL
  freopen("input.txt", "r", stdin);
#endif
  ios_base::sync_with_stdio(false);
  cin.tie(nullptr);

  ll n;
  cin >> n;

  n += 1;

  ll rest = n % mod;
  ll ans = fact(rest);

  ll k = n / mod;

  if (n >= mod * mod) {
    ++k;
    ans = ans * (mod - 1) % mod;
    ans = ans * fact(n / mod - mod) % mod;
  } else {
    ans = ans * fact(k) % mod;
  }

  ans = ans * binpow(mod - 1, k) % mod;

  cout << k << "␣" << ans;

  return 0;
}
```

Task E ()

```
#pragma GCC optimize("O3")


#include <algorithm>
#include <cassert>
#include <cmath>
#include <cstring>
#include <ctime>
#include <iomanip>
#include <iostream>
#include <map>
#include <numeric>
#include <queue>
#include <set>
#include <string>
#include <unordered_map>
#include <unordered_set>
#include <vector>

#define all(a) (a).begin(), (a).end()
#define ssize(a) (int)(a).size()

using namespace std;

using ll = long long;

const int max_n = 3e5 + 5;

int tree[max_n];
void update(int pos, int dlt) {
  for (int i = pos; i < max_n; i = i | (i + 1)) {
    tree[i] += dlt;
  }
}

int get(int r) {
  int res = 0;
  for (int i = r; i >= 0; i = (i & (i + 1)) - 1) {
    res += tree[i];
  }
  return res;
}

int get(int l, int r) { return get(r) - get(l - 1); }

struct Query {
  int l;
  int r;
  int d;
  int id;
};

bool operator<(const Query &lhs, const Query &rhs) {
  int bl = lhs.l >> 8;
  int br = rhs.l >> 8;
  if (bl == br) {
    if (bl & 1) {
      return lhs.r > rhs.r;
    }
    return lhs.r < rhs.r;
  }
  return bl < br;
}

int ceil_div(int a, int b) { return (a + b - 1) / b; }

int a[max_n];

int main() {
#ifdef LOCAL
  freopen("input.txt", "r", stdin);
#endif
```

```
ios_base::sync_with_stdio(false);
cin.tie(nullptr);

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

int q;
cin >> q;

vector<Query> queries(q);

for (int i = 0; i < q; ++i) {
  cin >> queries[i].l >> queries[i].r >> queries[i].d;
  --queries[i].r;
  queries[i].id = i;
}

sort(all(queries));

vector<ll> ans(q);

int l = 0;
int r = -1;

for (auto &el : queries) {
  while (l > el.l) {
    --l;
    update(a[l], 1);
  }
  while (r < el.r) {
    ++r;
    update(a[r], 1);
  }
  while (l < el.l) {
    update(a[l], -1);
    ++l;
  }
  while (r > el.r) {
    update(a[r], -1);
    --r;
  }

  int pref_cnt = 0;

  for (int i = 1; i <= el.d && i < max_n;) {
    int j = ceil_div(el.d, i);
    int next_i;
    if (j == 1) {
      next_i = max_n;
    } else {
      next_i = ceil_div(el.d, j - 1);
    }

    //cout << i << " " << next_i << " " << j << "\n";

    int ask_r = min(next_i - 1, max_n - 1);

    int new_pref_cnt = get(ask_r);
    int cnt = new_pref_cnt - pref_cnt;
    pref_cnt = new_pref_cnt;

    ans[el.id] += (ll)j * cnt;

    i = next_i;
  }
}

for (int i = 0; i < q; ++i) {
  cout << ans[i] << "\n";
}
```

```
  return 0;
}
```

Task F ()

```
#include <algorithm>
#include <cassert>
#include <cmath>
#include <cstring>
#include <ctime>
#include <iomanip>
#include <iostream>
#include <map>
#include <numeric>
#include <queue>
#include <set>
#include <string>
#include <unordered_map>
#include <unordered_set>
#include <vector>

#define all(a) (a).begin(), (a).end()
#define ssize(a) (int)(a).size()

using namespace std;

using ll = long long;


void compress(vector<ll>& a) {
  vector<ll> vals = a;
  sort(all(vals));
  vals.resize(unique(all(vals)) - vals.begin());
  for (auto &x : a) {
    x = lower_bound(all(vals), x) - vals.begin();
  }
}


struct BIT {
  vector<int> tree;

  BIT() {}

  BIT(int n) {
    tree.resize(n);
  }

  void update(int pos, int val) {
    for (int i = pos; i < ssize(tree); i = i | (i + 1)) {
      tree[i] = max(tree[i], val);
    }
  }

  int get(int r) {
    int res = 0;
    for (int i = r; i >= 0; i = (i & (i + 1)) - 1) {
      res = max(res, tree[i]);
    }
    return res;
  }
};

vector<BIT> tree;
vector<vector<int>> vals;

void update(int pos, int x, int y) {
  for (int i = pos; i >= 0; i = (i & (i + 1)) - 1) {
    int pos = lower_bound(all(vals[i]), x) - vals[i].begin();
    tree[i].update(pos, y);
  }
}

int get(int lx, int ry) {
  int res = 0;
  for (int i = lx; i < ssize(tree); i = i | (i + 1)) {
    int pos = upper_bound(all(vals[i]), ry) - vals[i].begin() - 1;
    if (pos >= 0) {
      res = max(res, tree[i].get(pos));
    }
  }
  return res;
}

int main() {
#ifdef LOCAL
  freopen("input.txt", "r", stdin);
#endif
  ios_base::sync_with_stdio(false);
  cin.tie(nullptr);

  int n;
  cin >> n;
  ++n;
  vector<int> l(n), r(n);
  for (int i = 1; i < n; ++i) {
    cin >> l[i] >> r[i];
  }

  vector<ll> pref_l(n), pref_r(n);
  for (int i = 1; i < n; ++i) {
    pref_l[i] = pref_l[i - 1] + l[i];
    pref_r[i] = pref_r[i - 1] + r[i];
  }

  compress(pref_l);
  compress(pref_r);

  tree.resize(ssize(pref_l));
  vals.resize(ssize(pref_l));

  for (int i = 0; i < n; ++i) {
    for (int j = pref_l[i]; j >= 0; j = (j & (j + 1)) - 1) {
      vals[j].push_back(pref_r[i]);
    }
  }
  for (int i = 0; i < n; ++i) {
    sort(all(vals[i]));
    vals[i].resize(unique(all(vals[i])) - vals[i].begin());
    tree[i] = BIT(ssize(vals[i]));
  }

  vector<int> dp(n);

  for (int i = 0; i < n; ++i) {
    int max_prev = get(pref_l[i], pref_r[i]);
    dp[i] = max_prev + 1;
    update(pref_l[i], pref_r[i], dp[i]);
  }

  int ans = *max_element(all(dp));

  cout << ans;

  return 0;
}
```