Олимпиада СПбГУ по информатике 2022/23 учебного года

A	B	C	D	E	F	Sum
100	100	80	10	16	65	371

Task A ()

```
a = [int(i) for i in input().split()]

x = []
for i, e in enumerate(a):
    e -= 1
    x = x[:e] + [i] + x[e:]

y = [x.index(i) + 1 for i in range(len(x))]
print(*y)
```

Task B ()

```
AAAA = 100000000


def first(n, a):
    print(sum(a) * AAAA)


def second(n, a):
    b = [i % AAAA for i in a]
    s = a[0] // AAAA
    print(sum(b) + s)


def main():
    w = input()
    n = int(input())
    a = list(map(int, input().split()))
    if w == 'first':
        first(n, a)
    else:
        second(n, a)


main()
```

Task C ()

```cpp
#pragma clang diagnostic push
#pragma ide diagnostic ignored "cppcoreguidelines-narrowing-conversions"
#pragma ide diagnostic ignored "OCUnusedTypeAliasInspection"
#pragma ide diagnostic ignored "OCUnusedGlobalDeclarationInspection"
#pragma ide diagnostic ignored "OCUnusedStructInspection"
#pragma ide diagnostic ignored "OCUnusedMacroInspection"
#ifndef DEBUG
//#pragma GCC optimize ("Ofast")
//#pragma GCC optimize ("unroll-loops")
//#pragma GCC target ("avx,avx2,fma")
#endif
#ifdef DEBUG
#define _GLIBCXX_DEBUG
#else
#define NDEBUG
#endif

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

#include <random>

using namespace std;

using ll = long long;
using ull = unsigned long long;
using ld = long double;
// using ii = ab<ll, ll>;
using ii = pair<ll, ll>;

using vb = vector<bool>;
using vi = vector<ll>;
using vii = vector<ii>;
using vc = vector<char>;

using vvi = vector<vi>;
using vvb = vector<vb>;
using vvc = vector<vc>;
using vsi = vector<set<ll>>;
using vmsi = vector<multiset<ll>>;
using vvii = vector<vii>;
template<typename T>
using vv = vector<vector<T>>;

using iii = tuple<ll, ll, ll>;
// using iiii = abcd<ll, ll, ll, ll>;
using v3i = vector<iii>;
// using v4i = vector<iiii>;

// const ll inf = 1e18;

#ifndef DEBUG
// interactive?
#define endl '\n'
#else
#endif

auto solve() -> void;

auto init() -> void;

extern bool enable_multitests;

auto main(int argc, char** argv) -> int {
#if defined(DEBUG)
  if (argc > 1 && string(argv[1]) == "-noredirect") {
    // ...
  } else {
    freopen("../in.txt", "r", stdin);
    freopen("../out.txt", "w", stdout);
  }
```

```
#endif

  ios_base::sync_with_stdio(false);
  cin.tie(nullptr);
  cout.tie(nullptr);
  cout.setf(ios::fixed);
  cout.precision(20);

  ll case_count = 1;
  ll case_id = 1;
  if (enable_multitests) {
    cin >> case_count;
  }
  init();
  while (case_count--) {
    solve();
    ++case_id;
  }
}

#ifndef DEBUG
mt19937_64 random_generator = mt19937_64(random_device()());
#else
mt19937_64 random_generator = mt19937_64(1337);  // NOLINT(cert-msc51-cpp,cert-err58-cpp)
#endif

inline ll rnd(ll a, ll b) {
  ll x = random_generator();
  x %= b - a + 1;
  x += a;
  return x;
}

extern const ll mod;

template<typename T>
T modpow(T base, T exp, T modulus) {
  base %= modulus;
  T result = 1;
  while (exp > 0) {
    if (exp & 1)
      result = (result * base) % modulus;
    base = (base * base) % modulus;
    exp >>= 1;
  }
  return result;
}

template<typename T>
T mod_invert(T a) {
  if (a < 1 or mod < 2)
    return -1;

  T u1 = mod;
  T u2 = 0;
  T v1 = a;
  T v2 = 1;

  while (v1 != 0) {
    T q = u1 / v1;
    T t1 = u1 - q * v1;
    T t2 = u2 - q * v2;
    u1 = v1;
    u2 = v2;
    v1 = t1;
    v2 = t2;
  }

  return u1 == 1 ? (u2 + mod) % mod : -1;
}

template<typename A, typename B>
ostream& operator<<(ostream& s, const pair<A, B>& self) {
  s << "{" << self.first << ", " << self.second << "}";
```

```
  return s;
}

template<typename T>
ostream& operator<<(ostream& s, const vector<T>& self) {
  for (auto e : self) {
    s << e << ' ';
  }
  return s;
}

template<typename T>
ostream& operator<<(ostream& s, const deque<T>& self) {
  for (auto e : self) {
    s << e << ' ';
  }
  return s;
}

template<typename A, typename B>
istream& operator>>(istream& s, pair<A, B>& self) {
  s >> self.first >> self.second;
  return s;
}

template<typename T>
istream& operator>>(istream& s, tuple<T, T, T>& self) {
  s >> get<0>(self) >> get<1>(self) >> get<2>(self);
  return s;
}

template<typename T>
istream& operator>>(istream& s, tuple<T, T, T, T>& self) {
  s >> get<0>(self) >> get<1>(self) >> get<2>(self) >> get<3>(self);
  return s;
}

template<typename T>
istream& operator>>(istream& s, vector<T>& self) {
  for (size_t i = 0; i < self.size(); ++i) {
    s >> self[i];
  }
  return s;
}


#if __cplusplus >= 201700L
template<typename Key, typename Value, typename Comp = less<>>
using ordered_map = __gnu_pbds::tree<Key,
                                     Value,
                                     Comp,
                                     __gnu_pbds::rb_tree_tag,
                                     __gnu_pbds::tree_order_statistics_node_update>;

template<typename T, typename Comp = less<>>
using ordered_set = ordered_map<T, __gnu_pbds::null_type, Comp>;

// template<typename T>
// inline void print_one(T arg) {
//   cout << arg << ' ';
// }
//
// template<typename... T>
// inline void print_(T&& ... args) {
//   (print_one(args), ...);
//   cout << endl;
// }

//#define print(...) print_(__VA_ARGS__)

#else
#define gcd __gcd
#endif  // __cplusplus
```

```
template<class T>
void prl(const T& t) {
  cout << t << endl;
}

auto factorize(ll n) -> map<ll, ll> {
  map<ll, ll> res;
  for (ll p = 2; n > 1 && p * p <= n; ++p) {
    while (n % p == 0) {
      n /= p;
      res[p] += 1;
    }
  }
  if (n > 1)
    res[n] += 1;
  return res;
}

bool is_prime_f(ll x) {
  for (int i = 2; i * i <= x; ++i) {
    if (x % i == 0)
      return false;
  }
  return true;
}

struct pair_hash {
  template<class T1, class T2>
  std::size_t operator()(const std::pair<T1, T2>& p) const {
    auto h1 = std::hash<T1>{}(p.first);
    auto h2 = std::hash<T2>{}(p.second);

    // Mainly for demonstration purposes, i.e. works but is overly simple
    // In the real world, use sth. like boost.hash_combine
    return h1 ^ h2;
  }
};

template<typename T>
auto sortpair(pair<T, T> p) -> pair<T, T> {
  if (p.second < p.first)
    return pair<T, T>{p.second, p.first};
  return p;
}

template<typename T>
auto flip(pair<T, T> p) -> pair<T, T> {
  return pair<T, T>{p.second, p.first};
}

#define MAP_ZERO_VAL_CHECK(map, key)       \
    {                                      \
        auto mapiter = (map).find((key)); \
        if (mapiter->second == 0) {        \
            (map).erase(mapiter);          \
        }                                  \
    }                                      \
    ((void)0)

#define RETURN_NO return prl("No")
#define RETURN_YES return prl("Yes")
#define RETURN_YESNO(...) {if (__VA_ARGS__) { RETURN_YES; } else { RETURN_NO; }} void(0)
#define RETURN(expr) return prl((expr))

#define X(...) __VA_ARGS__.first
#define Y(...) __VA_ARGS__.second

#define todo static_assert(false);

ll sqrtz(ll n) {
  ll L = -1, R = n + 1;
  while (R - L > 1) {
    ll M = (L + R) / 2;
    (M * M <= n ? L : R) = M;
```

```
  }
  return L;
}

bool issqr(ll n) {
  ll s = sqrtz(n);
  return s * s == n;
}

vi permute(ll n) {
  vi a(n);
  for (int i = 0; i < n; ++i) {
    a[i] = i;
  }
  return a;
}

//ll MOD(ll n) {
//  return ((n % mod) + mod) % mod;
//}

using __gnu_pbds::gp_hash_table;

// -----

constexpr ll module = 998244353;
//constexpr ll mod = 1e9 + 7;

// ----

auto init() -> void {}

// using i128 = __int128_t;

// ----

auto solve() -> void {
  ii a, b, c;
  cin >> a >> b >> c;

  set<ii> ans;
  ans.insert(sortpair(a));
  ans.insert(sortpair(b));
  ans.insert(sortpair(c));

  auto check3 = [&](ii a, ii b, ii c) {
    if (a.first == b.first && a.second > b.second) {
      ans.insert(sortpair(ii{a.first, a.second - b.second}));
    }
    if (b.first + c.first <= a.first && b.second + c.second > a.second && b.second < a.second && c
        .second <= a.second) {
      ans.insert(sortpair(ii{b.first, a.second - b.second}));
    }
    if (b.first + c.first == a.first && b.second + c.second == a.second) {
      ans.insert(sortpair(ii{b.first, c.second}));
    }
    if (b.first + c.first == a.first && b.second == c.second && b.second < a.second) {
      ans.insert(sortpair(ii{a.first, a.second - b.second}));
    }
    if (b.first <= a.first && c.first == a.first && b.second + c.second < a.second) {
      ans.insert(sortpair(ii{a.first, a.second - b.second - c.second}));
    }
  };

  vi p = permute(3);

  vii abc = {a, b, c};

  do {
    for (int i = 0; i < 8; ++i) {
      auto aa = abc[p[0]], bb = abc[p[1]], cc = abc[p[2]];
      if (i & 1) aa = flip(abc[p[0]]);
      if (i & 2) bb = flip(abc[p[1]]);
      if (i & 4) cc = flip(abc[p[2]]);
```

```
      check3(aa, bb, cc);
    }
  } while (next_permutation(p.begin(), p.end()));

  for (auto[x, y] : ans) {
    cout << x << ' ' << y << endl;
  }
}

bool enable_multitests = false;
#pragma clang diagnostic pop
```

Task D ()

```
#pragma clang diagnostic push
#pragma ide diagnostic ignored "cppcoreguidelines-narrowing-conversions"
#pragma ide diagnostic ignored "OCUnusedTypeAliasInspection"
#pragma ide diagnostic ignored "OCUnusedGlobalDeclarationInspection"
#pragma ide diagnostic ignored "OCUnusedStructInspection"
#pragma ide diagnostic ignored "OCUnusedMacroInspection"
#ifndef DEBUG
//#pragma GCC optimize ("Ofast")
//#pragma GCC optimize ("unroll-loops")
//#pragma GCC target ("avx,avx2,fma")
#endif
#ifdef DEBUG
#define _GLIBCXX_DEBUG
#else
#define NDEBUG
#endif

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

#include <random>

using namespace std;

using ll = long long;
using ull = unsigned long long;
using ld = long double;
// using ii = ab<ll, ll>;
using ii = pair<ll, ll>;

using vb = vector<bool>;
using vi = vector<ll>;
using vii = vector<ii>;
using vc = vector<char>;

using vvi = vector<vi>;
using vvb = vector<vb>;
using vvc = vector<vc>;
using vsi = vector<set<ll>>;
using vmsi = vector<multiset<ll>>;
using vvii = vector<vii>;
template<typename T>
using vv = vector<vector<T>>;

using iii = tuple<ll, ll, ll>;
// using iiii = abcd<ll, ll, ll, ll>;
using v3i = vector<iii>;
// using v4i = vector<iiii>;

// const ll inf = 1e18;

#ifndef DEBUG
// interactive?
//#define endl '\n'
#else
#endif

auto solve() -> void;

auto init() -> void;

extern bool enable_multitests;

auto main(int argc, char** argv) -> int {
#if defined(DEBUG)
  if (argc > 1 && string(argv[1]) == "-noredirect") {
    // ...
  } else {
    freopen("../in.txt", "r", stdin);
    freopen("../out.txt", "w", stdout);
  }
#endif
```

```
  ios_base::sync_with_stdio(false);
  cin.tie(nullptr);
  cout.tie(nullptr);
  cout.setf(ios::fixed);
  cout.precision(20);

  ll case_count = 1;
  ll case_id = 1;
  if (enable_multitests) {
    cin >> case_count;
  }
  init();
  while (case_count--) {
    solve();
    ++case_id;
  }
}

#ifndef DEBUG
mt19937_64 random_generator = mt19937_64(random_device()());
#else
mt19937_64 random_generator = mt19937_64(1337);  // NOLINT(cert-msc51-cpp,cert-err58-cpp)
#endif

inline ll rnd(ll a, ll b) {
  ll x = random_generator();
  x %= b - a + 1;
  x += a;
  return x;
}

extern const ll mod;

template<typename T>
T modpow(T base, T exp, T modulus) {
  base %= modulus;
  T result = 1;
  while (exp > 0) {
    if (exp & 1)
      result = (result * base) % modulus;
    base = (base * base) % modulus;
    exp >>= 1;
  }
  return result;
}

template<typename T>
T mod_invert(T a) {
  if (a < 1 or mod < 2)
    return -1;

  T u1 = mod;
  T u2 = 0;
  T v1 = a;
  T v2 = 1;

  while (v1 != 0) {
    T q = u1 / v1;
    T t1 = u1 - q * v1;
    T t2 = u2 - q * v2;
    u1 = v1;
    u2 = v2;
    v1 = t1;
    v2 = t2;
  }

  return u1 == 1 ? (u2 + mod) % mod : -1;
}

template<typename A, typename B>
ostream& operator<<(ostream& s, const pair<A, B>& self) {
  s << "{" << self.first << ", " << self.second << "}";
  return s;
```

```
}

template<typename T>
ostream& operator<<(ostream& s, const vector<T>& self) {
  for (auto e : self) {
    s << e << ' ';
  }
  return s;
}

template<typename T>
ostream& operator<<(ostream& s, const deque<T>& self) {
  for (auto e : self) {
    s << e << ' ';
  }
  return s;
}

template<typename A, typename B>
istream& operator>>(istream& s, pair<A, B>& self) {
  s >> self.first >> self.second;
  return s;
}

template<typename T>
istream& operator>>(istream& s, tuple<T, T, T>& self) {
  s >> get<0>(self) >> get<1>(self) >> get<2>(self);
  return s;
}

template<typename T>
istream& operator>>(istream& s, tuple<T, T, T, T>& self) {
  s >> get<0>(self) >> get<1>(self) >> get<2>(self) >> get<3>(self);
  return s;
}

template<typename T>
istream& operator>>(istream& s, vector<T>& self) {
  for (size_t i = 0; i < self.size(); ++i) {
    s >> self[i];
  }
  return s;
}

#if __cplusplus >= 201700L
template<typename Key, typename Value, typename Comp = less<>>
using ordered_map = __gnu_pbds::tree<Key,
                                     Value,
                                     Comp,
                                     __gnu_pbds::rb_tree_tag,
                                     __gnu_pbds::tree_order_statistics_node_update>;

template<typename T, typename Comp = less<>>
using ordered_set = ordered_map<T, __gnu_pbds::null_type, Comp>;

// template<typename T>
// inline void print_one(T arg) {
//   cout << arg << ' ';
// }
//
// template<typename... T>
// inline void print_(T&& ... args) {
//   (print_one(args), ...);
//   cout << endl;
// }

//#define print(...) print_(__VA_ARGS__)

#else
#define gcd __gcd
#endif  // __cplusplus

template<class T>
void prl(const T& t) {
```

```
  cout << t << endl;
}

auto factorize(ll n) -> map<ll, ll> {
  map<ll, ll> res;
  for (ll p = 2; n > 1 && p * p <= n; ++p) {
    while (n % p == 0) {
      n /= p;
      res[p] += 1;
    }
  }
  if (n > 1)
    res[n] += 1;
  return res;
}

bool is_prime_f(ll x) {
  for (int i = 2; i * i <= x; ++i) {
    if (x % i == 0)
      return false;
  }
  return true;
}

struct pair_hash {
  template<class T1, class T2>
  std::size_t operator()(const std::pair<T1, T2>& p) const {
    auto h1 = std::hash<T1>{}(p.first);
    auto h2 = std::hash<T2>{}(p.second);

    // Mainly for demonstration purposes, i.e. works but is overly simple
    // In the real world, use sth. like boost.hash_combine
    return h1 ^ h2;
  }
};

template<typename T>
auto sortpair(pair<T, T> p) -> pair<T, T> {
  if (p.second < p.first)
    return pair<T, T>{p.second, p.first};
  return p;
}

template<typename T>
auto flip(pair<T, T> p) -> pair<T, T> {
  return pair<T, T>{p.second, p.first};
}

#define MAP_ZERO_VAL_CHECK(map, key)       \
    {                                      \
        auto mapiter = (map).find((key)); \
        if (mapiter->second == 0) {        \
            (map).erase(mapiter);          \
        }                                  \
    }                                      \
    ((void)0)

#define RETURN_NO return prl("No")
#define RETURN_YES return prl("Yes")
#define RETURN_YESNO(...) {if (__VA_ARGS__) { RETURN_YES; } else { RETURN_NO; }} void(0)
#define RETURN(expr) return prl((expr))

#define X(...) __VA_ARGS__.first
#define Y(...) __VA_ARGS__.second

#define todo static_assert(false);

ll sqrtz(ll n) {
  ll L = -1, R = n + 1;
  while (R - L > 1) {
    ll M = (L + R) / 2;
    (M * M <= n ? L : R) = M;
  }
  return L;
```

```
}

bool issqr(ll n) {
  ll s = sqrtz(n);
  return s * s == n;
}

vi permute(ll n) {
  vi a(n);
  for (int i = 0; i < n; ++i) {
    a[i] = i;
  }
  return a;
}

//ll MOD(ll n) {
//  return ((n % mod) + mod) % mod;
//}

using __gnu_pbds::gp_hash_table;

// -----

constexpr ll module = 998244353;
//constexpr ll mod = 1e9 + 7;

// -----

auto init() -> void {}

// using i128 = __int128_t;

// -----

struct state : public _Resetiosflags {
  vi a;
  vb r;
  ull hash{};
  void recalc_hash() {
    hash = 0;
    for (int i = 0; i < a.size(); ++i) {
      hash += a[i];
      hash *= 1337;
      if (r[i]) {
        hash += 17;
        hash *= 7;
      }
    }
  }
  bool operator==(const state& other) const {
    return hash == other.hash;
  }
  bool operator<(const state& other) const {
//    if (a != other.a) return a < other.a;
//    return r < other.r;
    return hash < other.hash;
  }
};

struct state_hash {
  ull operator()(const state& s) const {
    return s.hash;
  }
};

ll n;
vi d;

gp_hash_table<state, bool, state_hash> st;
gp_hash_table<state, ii, state_hash> win_turn;

auto sim_move(const state& s) -> vector<ii> {
  vector<ii> ret;
  for (int i = 0; i < n; ++i) {
```

```
    for (int j = 0; j < s.a[i]; ++j) {
      if (j >= s.a[i] or j < 0) continue;
      ret.emplace_back(i, s.a[i] - j);
    }
    if (s.r[i]) {
      ret.emplace_back(i, 0);
    }
  }
  return ret;
}

auto solve_state(const state& s) -> bool {
  if (st.find(s) != st.end()) return st[s];
  {
    bool all_zeros = true;
    bool no_r = true;
    ll xa = 0;
    for (ll i = 0; i < s.a.size(); ++i) {
      xa ^= s.a[i];
      if (s.a[i] != 0) {
        all_zeros = false;
      }
      if (s.r[i]) {
        no_r = false;
      }
    }
    if (all_zeros) return st[s] = false;
    if (no_r && xa == 0) return st[s] = false;
  }
  for (const ii& turn2 : sim_move(s)) {
    auto s2 = s;
    if (turn2.second == 0) {
      s2.a[turn2.first] = d[turn2.first];
      s2.r[turn2.first] = false;
    } else {
      s2.a[turn2.first] -= turn2.second;
    }
    s2.recalc_hash();
    if (!solve_state(s2)) {
      win_turn[s] = turn2;
      return st[s] = true;
    }
  }
  return st[s] = false;
}

vi a;
vb r;

void apply(ll i, ll x) {
  if (i < 0 && x == -1) exit(0);
  else if (x == 0) r[i] = false, a[i] = d[i];
  else a[i] -= x;
}

auto turn(ll i, ll x) -> ii {
  if (i == -1 && x == -1) {
    cout << "-1 -1" << endl;
    exit(0);
  }
  apply(i, x);
  cout << i + 1 << ' ' << x << endl;
  cin >> i >> x;
  apply(i - 1, x);
  return {i - 1, x};
}

auto solve() -> void {
  cin >> n;
  a.assign(n, 0);
  cin >> a;
  d = a;
  r.assign(n, true);
```

```
#pragma clang diagnostic push
#pragma ide diagnostic ignored "EndlessLoop"
  while (true) {
    state s;
    s.a = a;
    s.r = r;
    s.recalc_hash();
    if (solve_state(s)) {
      turn(win_turn[s].first, win_turn[s].second);
    } else {
      turn(-1, -1);
    }
  }
#pragma clang diagnostic pop
}

bool enable_multitests = false;
#pragma clang diagnostic pop
```

Task E ()

```
#pragma clang diagnostic push
#pragma ide diagnostic ignored "cppcoreguidelines-narrowing-conversions"
#pragma ide diagnostic ignored "OCUnusedTypeAliasInspection"
#pragma ide diagnostic ignored "OCUnusedGlobalDeclarationInspection"
#pragma ide diagnostic ignored "OCUnusedStructInspection"
#pragma ide diagnostic ignored "OCUnusedMacroInspection"
#ifndef DEBUG
//#pragma GCC optimize ("Ofast")
//#pragma GCC optimize ("unroll-loops")
//#pragma GCC target ("avx,avx2,fma")
#endif
#ifdef DEBUG
#define _GLIBCXX_DEBUG
#else
#define NDEBUG
#endif

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

#include <random>

using namespace std;

using ll = long long;
using ull = unsigned long long;
using ld = long double;
// using ii = ab<ll, ll>;
using ii = pair<ll, ll>;

using vb = vector<bool>;
using vi = vector<ll>;
using vii = vector<ii>;
using vc = vector<char>;

using vvi = vector<vi>;
using vvb = vector<vb>;
using vvc = vector<vc>;
using vsi = vector<set<ll>>;
using vmsi = vector<multiset<ll>>;
using vvii = vector<vii>;
template<typename T>
using vv = vector<vector<T>>;

using iii = tuple<ll, ll, ll>;
// using iiii = abcd<ll, ll, ll, ll>;
using v3i = vector<iii>;
// using v4i = vector<iiii>;

// const ll inf = 1e18;

#ifndef DEBUG
// interactive?
//#define endl '\n'
#else
#endif

auto solve() -> void;

auto init() -> void;

extern bool enable_multitests;

string mode;

auto main(int argc, char** argv) -> int {
#if defined(DEBUG)
  if (argc > 1 && string(argv[1]) == "-noredirect") {
    // ...
  } else {
    freopen("../in.txt", "r", stdin);
    freopen("../out.txt", "w", stdout);
```

```
  }
#endif

  ios_base::sync_with_stdio(false);
  cin.tie(nullptr);
  cout.tie(nullptr);
  cout.setf(ios::fixed);
  cout.precision(20);

  init();

  ll t;
  cin >> t;
  cin >> mode;
  while (t--) solve();
}

#ifndef DEBUG
mt19937_64 random_generator = mt19937_64(random_device()());
#else
mt19937_64 random_generator = mt19937_64(1337);  // NOLINT(cert-msc51-cpp,cert-err58-cpp)
#endif

inline ll rnd(ll a, ll b) {
  ll x = random_generator();
  x %= b - a + 1;
  x += a;
  return x;
}

extern const ll mod;

template<typename T>
T modpow(T base, T exp, T modulus) {
  base %= modulus;
  T result = 1;
  while (exp > 0) {
    if (exp & 1)
      result = (result * base) % modulus;
    base = (base * base) % modulus;
    exp >>= 1;
  }
  return result;
}

template<typename T>
T mod_invert(T a) {
  if (a < 1 or mod < 2)
    return -1;

  T u1 = mod;
  T u2 = 0;
  T v1 = a;
  T v2 = 1;

  while (v1 != 0) {
    T q = u1 / v1;
    T t1 = u1 - q * v1;
    T t2 = u2 - q * v2;
    u1 = v1;
    u2 = v2;
    v1 = t1;
    v2 = t2;
  }

  return u1 == 1 ? (u2 + mod) % mod : -1;
}

template<typename A, typename B>
ostream& operator<<(ostream& s, const pair<A, B>& self) {
  s << "{" << self.first << ", " << self.second << "}";
  return s;
}
```

```cpp
template<typename T>
ostream& operator<<(ostream& s, const vector<T>& self) {
  for (auto e : self) {
    s << e << ' ';
  }
  return s;
}

template<typename T>
ostream& operator<<(ostream& s, const deque<T>& self) {
  for (auto e : self) {
    s << e << ' ';
  }
  return s;
}

template<typename A, typename B>
istream& operator>>(istream& s, pair<A, B>& self) {
  s >> self.first >> self.second;
  return s;
}

template<typename T>
istream& operator>>(istream& s, tuple<T, T, T>& self) {
  s >> get<0>(self) >> get<1>(self) >> get<2>(self);
  return s;
}

template<typename T>
istream& operator>>(istream& s, tuple<T, T, T, T>& self) {
  s >> get<0>(self) >> get<1>(self) >> get<2>(self) >> get<3>(self);
  return s;
}

template<typename T>
istream& operator>>(istream& s, vector<T>& self) {
  for (size_t i = 0; i < self.size(); ++i) {
    s >> self[i];
  }
  return s;
}


#if __cplusplus >= 201700L
template<typename Key, typename Value, typename Comp = less<>>
using ordered_map = __gnu_pbds::tree<Key,
                                     Value,
                                     Comp,
                                     __gnu_pbds::rb_tree_tag,
                                     __gnu_pbds::tree_order_statistics_node_update>;

template<typename T, typename Comp = less<>>
using ordered_set = ordered_map<T, __gnu_pbds::null_type, Comp>;

// template<typename T>
// inline void print_one(T arg) {
//   cout << arg << ' ';
// }
//
// template<typename... T>
// inline void print_(T&& ... args) {
//   (print_one(args), ...);
//   cout << endl;
// }

//#define print(...) print_(__VA_ARGS__)

#else
#define gcd __gcd
#endif  // __cplusplus

template<class T>
void prl(const T& t) {
  cout << t << endl;
```

```
}

auto factorize(ll n) -> map<ll, ll> {
  map<ll, ll> res;
  for (ll p = 2; n > 1 && p * p <= n; ++p) {
    while (n % p == 0) {
      n /= p;
      res[p] += 1;
    }
  }
  if (n > 1)
    res[n] += 1;
  return res;
}

bool is_prime_f(ll x) {
  for (int i = 2; i * i <= x; ++i) {
    if (x % i == 0)
      return false;
  }
  return true;
}

struct pair_hash {
  template<class T1, class T2>
  std::size_t operator()(const std::pair<T1, T2>& p) const {
    auto h1 = std::hash<T1>{}(p.first);
    auto h2 = std::hash<T2>{}(p.second);

    // Mainly for demonstration purposes, i.e. works but is overly simple
    // In the real world, use sth. like boost.hash_combine
    return h1 ^ h2;
  }
};

template<typename T>
auto sortpair(pair<T, T> p) -> pair<T, T> {
  if (p.second < p.first)
    return pair<T, T>{p.second, p.first};
  return p;
}

template<typename T>
auto flip(pair<T, T> p) -> pair<T, T> {
  return pair<T, T>{p.second, p.first};
}

#define MAP_ZERO_VAL_CHECK(map, key)       \
    {                                      \
        auto mapiter = (map).find((key)); \
        if (mapiter->second == 0) {        \
            (map).erase(mapiter);          \
        }                                  \
    }                                      \
    ((void)0)

#define RETURN_NO return prl("No")
#define RETURN_YES return prl("Yes")
#define RETURN_YESNO(...) {if (__VA_ARGS__) { RETURN_YES; } else { RETURN_NO; }} void(0)
#define RETURN(expr) return prl((expr))

#define X(...) __VA_ARGS__.first
#define Y(...) __VA_ARGS__.second

#define todo static_assert(false);

ll sqrtz(ll n) {
  ll L = -1, R = n + 1;
  while (R - L > 1) {
    ll M = (L + R) / 2;
    (M * M <= n ? L : R) = M;
  }
  return L;
}
```

```cpp
bool issqr(ll n) {
  ll s = sqrtz(n);
  return s * s == n;
}

vi permute(ll n) {
  vi a(n);
  for (int i = 0; i < n; ++i) {
    a[i] = i;
  }
  return a;
}

//ll MOD(ll n) {
//  return ((n % mod) + mod) % mod;
//}

using __gnu_pbds::gp_hash_table;

// -----

constexpr ll module = 998244353;
//constexpr ll mod = 1e9 + 7;

// ----

vvi mp;

auto init() -> void {
  for (int a0 = 0; a0 <= 10; ++a0) {
    for (int a1 = a0; a1 <= 10; ++a1) {
      for (int a2 = a1; a2 <= 10; ++a2) {
        for (int a3 = a2; a3 <= 10; ++a3) {
          for (int a4 = a3; a4 <= 10; ++a4) {
            for (int a5 = a4; a5 <= 10; ++a5) {
              for (int a6 = a5; a6 <= 10; ++a6) {
                for (int a7 = a6; a7 <= 10; ++a7) {
                  for (int a8 = a7; a8 <= 10; ++a8) {
                    for (int a9 = a8; a9 <= 10; ++a9) {
                      mp.emplace_back(vi{a0, a1, a2, a3, a4, a5, a6, a7, a8, a9});
                    }
                  }
                }
              }
            }
          }
        }
      }
    }
  }
}

// using i128 = __int128_t;

// ----

auto solve() -> void {
  if (mode == "transmit") {
    ll n;
    cin >> n;

    vvc a(10, vc(10, '0'));

    auto val = mp[n];
    for (int i = 0; i < 10; ++i) {
      for (int j = 0; j < val[i]; ++j) {
        a[i][j] = '1';
      }
    }

    for (const auto& line : a) {
      for (const auto& c : line) {
        cout << c;
```

```
      }
      cout << endl;
    }
    cout << endl;
  } else {
    vvc a(10, vc(10));
    cin >> a;
    vi val(10);
    for (int i = 0; i < 10; ++i) {
      val[i] = count(a[i].begin(), a[i].end(), '1');
    }
    sort(val.begin(), val.end());
    cout << lower_bound(mp.begin(), mp.end(), val) - mp.begin() << endl;
  }
}

bool enable_multitests = false;
#pragma clang diagnostic pop
```

Task F ()

```
def unzip(x):
    r = ''
    i = 0
    while i < len(x):
        if x[i] == '(':
            j = x[i:].index(')')
            a, b = x[i + 1:j].split('|')
            b = int(b)
            r += a * b
            i = j + 1
        elif '(' in x[i:]:
            j = i + x[i:].index('(')
            r += x[i:j]
            i = j
        else:
            r += x[i:]
            i = len(x)
    return r


def main():
    a = int(unzip(input()))
    b = int(unzip(input()))
    print(a + b)


main()
```