{

    "": {

        "prefix": "all\_pre\_built\_fnc",

        "body": [

          "ll binpow(ll a,ll b) {",

          "    ll ans = 1;",

          "    while(b > 0) {",

          "        if((b & 1) == 1) ans \*= a;",

          "        a \*= a;",

          "        b = b >> 1;",

          "    }",

          "    return ans;",

          "}",

          "",

          "ll binpowmod(ll a,ll b) {",

          "    ll ans = 1;",

          "    while(b > 0) {",

          "        if((b & 1) == 1) {",

          "         ans \*= a;",

          "         ans %= mod;",

          "        }",

          "        a \*= a;",

          "        a %= mod;",

          "        b = b >> 1;",

          "    }",

          "    return ans;",

          "}",

          "",

          "",

          "ll gcd(ll a,ll b) {",

          " if(b == 0) return a;",

          " return gcd(b, a % b);",

          "}",

          "",

          "ll lcm(ll a,ll b) {",

          " return (a / gcd(a,b)) \* b;",

          "}",

          "",

          "const ll MAX = 2e5 + 7;",

          "vector<ll> fact(MAX);",

          " ",

          "ll add(ll a, ll b) {",

          " return (a + b) % mod;",

          "}",

          " ",

          "ll sub(ll a, ll b) {",

          " return ((a - b) % mod + mod) % mod;",

          "}",

          " ",

          "ll mult(ll a, ll b) {",

          " return ((a \* b) % mod);",

          "}",

          "",

          "ll inv(ll a) {",

          " return binpowmod(a, mod - 2);",

          "}",

          " ",

          "ll divide(ll a, ll b) {",

          " return mult(a, inv(b));",

          "}",

          " ",

          "ll nCr(ll n, ll r) {",

          " if(n < r) return 0;",

          " return divide(fact[n], mult(fact[r], fact[n - r]));",

          "}",

          " ",

          "void preFactorial() {",

          " fact[0] = 1;",

          " for(ll i = 1; i < MAX; i++) fact[i] = mult(i, fact[i - 1]);",

          "}",

          "",

          "bool isVowel(char c) {",

          " if(c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u') return true;",

          " if(c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U') return true;",

          " return false;",

          "}",

          "",

          "bool isSame(ll n, ll arr[]) {",

          " for(ll i = 0; i < n; i++) {",

          "     if(arr[i] != arr[0]) return false;",

          " }",

          " return true;",

          "}",

          "",

          "bool isSame(ll n, vector<ll> &arr) {",

          " for(ll i = 0; i < n; i++) {",

          "     if(arr[i] != arr[0]) return false;",

          " }",

          " return true;",

          "}",

          "",

          "bool isSorted(ll n, ll arr[]) {",

          " for(ll i = 1; i < n; i++) {",

          "     if(arr[i] < arr[i - 1]) return false;",

          " }",

          " return true;",

          "}",

          "",

          "bool isSorted(ll n, vector<ll> &arr) {",

          " for(ll i = 1; i < n; i++) {",

          "     if(arr[i] < arr[i - 1]) return false;",

          " }",

          " return true;",

          "}",

          "",

          "void inputArr(ll n, ll arr[]) {",

          " for(ll i = 0; i < n; i++) cin >> arr[i];",

          "}",

          "",

          "void inputArr(ll n, vector<ll> &arr) {",

          " ll x;",

          " for(ll i = 0; i < n; i++) {",

          "     cin >> x;",

          "     arr.push\_back(x);",

          " }",

          "}",

          "",

          "void printArr(ll n, ll arr[]) {",

          " for(ll i = 0; i < n; i++) cout << arr[i] << \" \";",

          " cout << nl;",

          "}",

          "",

          "void printArr(ll n, vector<ll> &arr) {",

          " for(ll i = 0; i < n; i++) cout << arr[i] << \" \";",

          " cout << nl;",

          "}",

          "",

          "ll sumOfArr(ll n, ll arr[]) {",

          " ll ans = 0;",

          " for(ll i = 0; i < n; i++) ans += arr[i];",

          " return ans;",

          "}",

          "",

          "ll sumOfArr(ll n, vector<ll> &arr) {",

          " ll ans = 0;",

          " for(ll i = 0; i < n; i++) ans += arr[i];",

          " return ans;",

          "}",

          "",

          "bool isPrime(ll n) {",

          " if(n == 1) return false;",

          " for(ll i = 2; i <= sqrt(n); i++) {",

          "     if(n % i == 0) return false;",

          " }",

          " return true;",

          "}",

          "",

          "ll countSetBits(ll n) {",

          " ll ans = 0;",

          " while(n) {",

          "     ans++;",

          "     n = n & (n - 1);",

          " }",

          " return ans;",

          "}",

          "",

          "vector<ll> primeFactorization(ll n) {",

          " vector<ll> factors;",

          " for(ll i = 2; i \* i <= n; i++) {",

          "     ll cnt = 0; ",

          "     while(n % i == 0) {",

          "         cnt++;",

          "         n /= i;",

          "         factors.push\_back(i);",

          "     }",

          " }",

          " if(n > 1) factors.push\_back(n);",

          " return factors;",

          "}",

          "",

          "bool isPalindrome(string s) {",

          " ll i = 0;",

          " ll j = s.size() - 1;",

          " while(i <= j) {",

          "     if(s[i] != s[j]) return false;",

          "     i++;",

          "     j--;",

          " }",

          " return true;",

          "}"

        ],

        "description": ""

      }

}