```
1: ################################################
2: ################# ./template.cpp #################
3: ################################################
4:
5: #include <bits/stdc++.h>
6: using namespace std;
7: typedef long long ll;
8: typedef vector<int> vint;
9: typedef pair<int,int> pint;
10: typedef vector<pint> vpint;
11: #define rep(i,n) for(int i=0;i<(n);i++)
12: #define REP(i,n) for(int i=n-1;i>=(0);i--)
13: #define reps(i,f,n) for(int i=(f);i<(n);i++)
14: #define each(it,v) for(__typeof((v).begin()) it=(v).begin();it!=(v).end();it++)
15: #define all(v) (v).begin(),(v).end()
16: #define eall(v) unique(all(v), v.end())
17: #define pb push_back
18: #define mp make_pair
19: #define fi first
20: #define se second
21: #define chmax(a, b) a = (((a)<(b)) ? (b) : (a))
22: #define chmin(a, b) a = (((a)>(b)) ? (b) : (a))
23: const int MOD = 1e9 + 7;
24: const int INF = 1e9;
25: const ll INFF = 1e18;
26:
27: int main(void){
28:         return 0;
29: }
30:
31: ################################################
32: ########### ./data_struture/imos_one.h ###########
33: ################################################
34:
35: vector<int> imos_one(vector<int> baseimos){
36:     int basesize = baseimos.size();
37:     vector<int> imos(basesize, 0);
38:     imos[0] = baseimos[0];
39:     for(int i = 1; i < basesize; ++i){
40:         imos[i] = imos[i - 1] + baseimos[i];
41:     }
42:     return imos;
43: }
44:
45: ################################################
46: #### ./data_struture/segtree_lazy_add_min.cpp ####
47: ################################################
48:
49: //(1)å\214ºé\226\223ã\201«ä¸ \200æ§\230å\212 ç®\227 (2)å\214ºé\226\223ã\201®æ\234\200å°\217å\200¤
50: struct segtree{
51: public:
52:     const int SIZE = 1 << 18;
53:     //seg:å\214ºé\226\223ã\201®æ\234\200å°\217å\200¤ lazy:å\214ºé\226\223ã\201«å¯¾ã\201\227ã\201¦ã\200\201å\212 ã\201\210ã\202\213å\200¤ã\201§ã\201¾ã\201 é\201\205å»¶ã\201\227ã\201¦ã\201\204ã\202\213ã\202\202ã\201®
54:     vector<ll> seg, lazy;//segã\201¯æ¬²ã\201\227ã\201\204æ\203\205å ± lazyã\201¯å\214ºé\226\223ã\201«å¯¾ã\201\231ã\202\213ä¸ \200æ§\230ã\201ªå\207¦ç\220\206ã\202\222ç¤ºã\201\231ã\202\202ã\201®
55:     segtree():seg(SIZE * 2), lazy(SIZE * 2){}
56:     void lazy_evaluate(int k, int l, int r){//é\201\205å»¶æ\203\205å ±ã\201®é\201©ç\224¨æ\226¹æ³\225
57:         if(lazy[k] != 0){
58:             seg[k] += lazy[k];//å\214ºé\226\223[l,r)ã\201«ã\201\231ã\201¹ã\201¦å\220\214ã\201\230å\200¤ã\202\222è¿½å\212 ã\201\231ã\202\213ã\201\223ã\201¨ã\201«ã\201ªã\201£ã\201¦ã\201\204ã\201¦ã\200\201segã\201«ã\201¯æ\234\200å°\217å\200¤ã\201\214å\205¥ã\201£ã\201¦ã\201\204ã\202\213ã\201®ã\201§ã\200\201å\212 ã\201\210ã\202\213å\200¤ã\202\222è¶³ã\201\231
59:             if(r  - l > 1){
60:                 lazy[k * 2 + 1] += lazy[k];//é\201\205å»¶ã\202\222å•¦ã\201®å-\220ã\201«ä¼\235æ\220¬
61:                 lazy[k * 2 + 2] += lazy[k];//é\201\205å»¶ã\202\222å\217³ã\201®å-\220ã\201«ä¼\235æ\220¬
62:             }
63:             lazy[k] = 0;//ã\203\216ã\203¼ã\203\211kã\201¯ä¼\235æ\220¬å®\214äº\206
64:         }
65:     }
66:     void update(int a, int b, int k, int l, int r, int x){
67:         lazy_evaluate(k, l, r);
68:         if(r <= a || b <= l) return;
69:         if(a <= l && r <= b){
70:             lazy[k] += x; //å\212 ã\201\210ã\202\213
71:             lazy_evaluate(k, l, r);
72:         }else{
73:             update(a, b, k * 2 + 1, l, (l + r) / 2, x);
74:             update(a, b, k * 2 + 2, (l + r) / 2, r, x);
75:             seg[k] = min(seg[k * 2 + 1], seg[k * 2 + 2]); //å\214ºé\226\223ã\201®min
76:         }
77:     }
78:     ll query(int a, int b, int k, int l, int r){
79:         lazy_evaluate(k, l, r);
80:         if(r <= a || b <= l) return INF;//minã\201®å½±é\237¿ã\201®ã\201ªã\201\204ã\202\202ã\201®
81:         if(a <= l && r <= b) return seg[k];
82:         ll x = query(a, b, k * 2 + 1, l, (l + r) / 2);
83:         ll y = query(a, b, k * 2 + 2, (l + r) / 2, r);
84:         return min(x, y); //å•¦å\217³ã\201®minã\202\222
85:     }
86:     //update(a,b,x) := [a,b)ã\202\222å\205¨ã\201¦xã\202\222å\212 ã\201\210ã\202\213
87:     void update(int a, int b, int x){update(a, b, 0, 0, SIZE, x);}
88:     //query(a,b) := [a,b)ã\201«å¯¾ã\201\231ã\202\213æ\234\200å°\217å\200¤ã\202\222æ±\202ã\202\201ã\202\213
89:     ll query(int a, int b){return query(a, b, 0, 0, SIZE);}
90: };
91:
92: /*
93: //(1)å\214ºé\226\223ã\201«ä¸ \200æ§\230å\212 ç®\227 (2)å\214ºé\226\223ã\201®æ\234\200å°\217å\200¤
94: const int MAX_N = 1 << 19;
95: struct segtree{
96: public:
97:         int SIZE;
98:         int seg[MAX_N * 2], lazy[MAX_N * 2];//segã\201¯æ¬²ã\201\227ã\201\204æ\203\205å ± lazyã\201¯å\214ºé\226\223ã\201«å¯¾ã\201\231ã\202\213ä¸ \200æ§\230ã\201ªå\207¦ç\220\206ã\202\222ç¤ºã\201\231ã\202\202ã\201®
99:         //seg:å\214ºé\226\223ã\201®æ\234\200å°\217å\200¤ lazy:å\214ºé\226\223ã\201«å¯¾ã\201\227ã\201¦ã\200\201å\212 ã\201\210ã\202\213å\200¤ã\201§ã\201¾ã\201 é\201\205å»¶ã\201\227ã\201¦ã\201\204ã\202\213ã\202\202ã\201®
100:     segtree(int n){
101:         SIZE = 1;
102:         while(SIZE < n) SIZE *= 2;//è¦\201ç´ æ\225°ã\202\2222ã\201®ã\201¹ã\201\215ä¹\227ã\201«
103:         for (int i = 0; i < 2 * SIZE - 1; ++i) seg[i] = lazy[i] = 0;//0ã\201¯é\201\205å»¶ã\201\227ã\201¦ã\201\204ã\202\213ã\202\202ã\201®ã\201\214ã\201ªã\201\204ã\201\223ã\201¨ã\202\222ç¤ºã\201\231
104:     }
105:     void lazy_evaluate(int k, int l, int r){//é\201\205å»¶æ\203\205å ±ã\201®é\201©ç\224¨æ\226¹æ³\225
106:         if(lazy[k] != 0){
107:             seg[k] += lazy[k];//å\214ºé\226\223[l,r)ã\201«ã\201\231ã\201¹ã\201¦å\220\214ã\201\230å\200¤ã\202\222è¿½å\212 ã\201\231ã\202\213ã\201\223ã\201¨ã\201«ã\201ªã\201£ã\201¦ã\201\204ã\201¦ã\200\201segã\201«ã\201¯æ\234\200å°\217å\200¤ã\201\214å\205¥ã\201£ã\201¦ã\201\204ã\202\213ã\201®ã\201§ã\200\201å\212 ã\201\210ã\202\213å\200¤ã\202\222è¶³ã\201\231
```

```
            if(r  - l > 1){
                lazy[k * 2 + 1] += lazy[k];//é\201\205å»¶ã\202\222å•¦ã\201®å-\220ã\201«ä¼\235æ\220¬
                lazy[k * 2 + 2] += lazy[k];//é\201\205å»¶ã\202\222å\217³ã\201®å-\220ã\201«ä¼\235æ\220¬
            }
            lazy[k] = 0;//ã\203\216ã\203¼ã\203\211kã\201¯ä¼\235æ\220¬å®\214äº\206
        }
    }
    void update(int a, int b, int k, int l, int r, int x){
        lazy_evaluate(k, l, r);
        if(r <= a || b <= l) return;
        if(a <= l && r <= b){
            lazy[k] += x; //å\212 ã\201\210ã\202\213
            lazy_evaluate(k, l, r);
        }else{
            update(a, b, k * 2 + 1, l, (l + r) / 2, x);
            update(a, b, k * 2 + 2, (l + r) / 2, r, x);
            seg[k] = min(seg[k * 2 + 1], seg[k * 2 + 2]); //å\214ºé\226\223ã\201®min
        }
    }
    ll query(int a, int b, int k, int l, int r){
        lazy_evaluate(k, l, r);
        if(r <= a || b <= l) return INF;//minã\201®å½±é\237¿ã\201®ã\201ªã\201\204ã\202\202ã\201®
        if(a <= l && r <= b) return seg[k];
        ll x = query(a, b, k * 2 + 1, l, (l + r) / 2);
        ll y = query(a, b, k * 2 + 2, (l + r) / 2, r);
        return min(x, y); //å•¦å\217³ã\201®minã\202\222
    }
    //update(a,b,x) := [a,b)ã\202\222å\205¨ã\201¦xã\202\222å\212 ã\201\210ã\202\213
    void update(int a, int b, int x){update(a, b, 0, 0, SIZE, x);}
    //query(a,b) := [a,b)ã\201«å¯¾ã\201\231ã\202\213æ\234\200å°\217å\200¤ã\202\222æ±\202ã\202\201ã\202\213
    ll query(int a, int b){return query(a, b, 0, 0, SIZE);}
};
*/

/*
http://tenka1-2016-qualb.contest.atcoder.jp/tasks/tenka1_2016_qualB_d
*/


////////////////////////////////////////////////////
///// ./data_struture/segtree_lazy_add_min.h /////
////////////////////////////////////////////////////

//(1)å\214ºé\226\223ã\201«ä¸\200æ§\230å\212 ç®\227 (2)å\214ºé\226\223ã\201®æ\234\200å°\217å\200¤
struct segtree{
public:
    const int SIZE = 1 << 18;
    //seg:å\214ºé\226\223ã\201®æ\234\200å°\217å\200¤ lazy:å\214ºé\226\223ã\201«å¯¾ã\201\227ã\201¦ã\200\201å\212 ã\201\210ã\202\213å\200¤ã\201§ã\201¾ã\201 é\201\205å»¶ã\201\227ã\201¦ã\201\204ã\202\213ã\202\202ã\201®
    vector<ll> seg, lazy;//segã\201¯æ¬²ã\201\227ã\201\204æ\203\205å ± lazyã\201¯å\214ºé\226\223ã\201«å¯¾ã\201\231ã\202\213ä¸\200æ§\230ã\201ªå\207¦ç\220\206ã\202\222ç¤ºã\201\231ã\202\202ã\201®
    segtree():seg(SIZE * 2), lazy(SIZE * 2){}
    void lazy_evaluate(int k, int l, int r){//é\201\205å»¶æ\203\205å ±ã\201®é\201©ç\224¨æ\226¹æ³\225
        if(lazy[k] != 0){
            seg[k] += lazy[k];//å\214ºé\226\223[l,r)ã\201«ã\201\231ã\201¹ã\201¦å\220\214ã\201\230å\200¤ã\202\222è¿½å\212 ã\201\231ã\202\213ã\201\223ã\201¨ã\201«ã\201ªã\201£ã\201¦ã\201\204ã\201¦ã\200\201segã\201«ã\201¯æ\234\200å°\217å\200¤ã\201\214å\205¥ã\201£ã\201¦ã\201\204ã\202\213ã\201®ã\201§ã\200\201å\212 ã\201\210ã\202\213å\200¤ã\202\222è¶³ã\201\231
        }
        lazy[k] = 0;//ã\203\216ã\203¼ã\203\211kã\201¯ä¼\235æ\220¬å®\214äº\206
    }
    void update(int a, int b, int k, int l, int r, int x){
        lazy_evaluate(k, l, r);
        if(r <= a || b <= l) return;
        if(a <= l && r <= b){
            lazy[k] += x; //å\212 ã\201\210ã\202\213
            lazy_evaluate(k, l, r);
        }else{
            update(a, b, k * 2 + 1, l, (l + r) / 2, x);
            update(a, b, k * 2 + 2, (l + r) / 2, r, x);
            seg[k] = min(seg[k * 2 + 1], seg[k * 2 + 2]); //å\214ºé\226\223ã\201®min
        }
    }
    ll query(int a, int b, int k, int l, int r){
        lazy_evaluate(k, l, r);
        if(r <= a || b <= l) return INF;//minã\201®å½±é\237¿ã\201®ã\201ªã\201\204ã\202\202ã\201®
        if(a <= l && r <= b) return seg[k];
        ll x = query(a, b, k * 2 + 1, l, (l + r) / 2);
        ll y = query(a, b, k * 2 + 2, (l + r) / 2, r);
        return min(x, y); //å•¦å\217³ã\201®minã\202\222
    }
    //update(a,b,x) := [a,b)ã\202\222å\205¨ã\201¦xã\202\222å\212 ã\201\210ã\202\213
    void update(int a, int b, int x){update(a, b, 0, 0, SIZE, x);}
    //query(a,b) := [a,b)ã\201«å¯¾ã\201\231ã\202\213æ\234\200å°\217å\200¤ã\202\222æ±\202ã\202\201ã\202\213
    ll query(int a, int b){return query(a, b, 0, 0, SIZE);}
};

/*
//(1)å\214ºé\226\223ã\201«ä¸\200æ§\230å\212 ç®\227 (2)å\214ºé\226\223ã\201®æ\234\200å°\217å\200¤
const int MAX_N = 1 << 19;
struct segtree{
public:
    int SIZE;
    int seg[MAX_N * 2], lazy[MAX_N * 2];//segã\201¯æ¬²ã\201\227ã\201\204æ\203\205å ± lazyã\201¯å\214ºé\226\223ã\201«å¯¾ã\201\231ã\202\213ä¸\200æ§\230ã\201ªå\207¦ç\220\206ã\202\222ç¤ºã\201\231ã\202\202ã\201®
    //seg:å\214ºé\226\223ã\201®æ\234\200å°\217å\200¤ lazy:å\214ºé\226\223ã\201«å¯¾ã\201\227ã\201¦ã\200\201å\212 ã\201\210ã\202\213å\200¤ã\201§ã\201¾ã\201 é\201\205å»¶ã\201\227ã\201¦ã\201\204ã\202\213ã\202\202ã\201®
    segtree(int n){
        SIZE = 1;
        while(SIZE < n) SIZE *= 2;//è¦\201ç´ æ\225°ã\202\2222ã\201®ã\201¹ã\201\215ä¹\227ã\201«
        for (int i = 0; i < 2 * SIZE - 1; ++i) seg[i] = lazy[i] = 0;//0ã\201¯é\201\205å»¶ã\201\227ã\201¦ã\201\204ã\202\213ã\202\202ã\201®ã\201\214ã\201ªã\201\204ã\201\223ã\201¨ã\202\222ç¤ºã\201\231
    }
    void lazy_evaluate(int k, int l, int r){//é\201\205å»¶æ\203\205å ±ã\201®é\201©ç\224¨æ\226¹æ³\225
        if(lazy[k] != 0){
            seg[k] += lazy[k];//å\214ºé\226\223[l,r)ã\201«ã\201\231ã\201¹ã\201¦å\220\214ã\201\230å\200¤ã\202\222è¿½å\212 ã\201\231ã\202\213ã\201\223ã\201¨ã\201«ã\201ªã\201£ã\201¦ã\201\204ã\201¦ã\200\201segã\201«ã\201¯æ\234\200å°\217å\200¤ã\201\214å\205¥ã\201£ã\201¦ã\201\204ã\202\213ã\201®ã\201§ã\200\201å\212 ã\201\210ã\202\213å\200¤ã\202\222è¶³ã\201\231
            if(r  - l > 1){
```

```
                lazy[k * 2 + 1] += lazy[k];//é\201\205å»¶ã\202\222å•¦ã\201®å-\220ã\201«ä¼\235æ\220¬
                lazy[k * 2 + 2] += lazy[k];//é\201\205å»¶ã\202\222å\217³ã\201®å-\220ã\201«ä¼\235æ\220¬
            }
            lazy[k] = 0;//ã\203\216ã\203¼ã\203\211kã\201¯ä¼\235æ\220¬å®\214äº\206
        }
    }
    void update(int a, int b, int k, int l, int r, int x){
        lazy_evaluate(k, l, r);
        if(r <= a || b <= l) return;
        if(a <= l && r <= b){
            lazy[k] += x; //å\212 ã\201\210ã\202\213
            lazy_evaluate(k, l, r);
        }else{
            update(a, b, k * 2 + 1, l, (l + r) / 2, x);
            update(a, b, k * 2 + 2, (l + r) / 2, r, x);
            seg[k] = min(seg[k * 2 + 1], seg[k * 2 + 2]); //å\214ºé\226\223ã\201®min
        }
    }
    ll query(int a, int b, int k, int l, int r){
        lazy_evaluate(k, l, r);
        if(r <= a || b <= l) return INF;//minã\201®å½±é\237¿ã\201®ã\201ªã\201\204ã\202\202ã\201®
        if(a <= l && r <= b) return seg[k];
        ll x = query(a, b, k * 2 + 1, l, (l + r) / 2);
        ll y = query(a, b, k * 2 + 2, (l + r) / 2, r);
        return min(x, y); //å•¦å\217³ã\201®minã\202\222
    }
    //update(a,b,x) := [a,b)ã\202\222å\205¨ã\201¦xã\202\222å\212 ã\201\210ã\202\213
    void update(int a, int b, int x){update(a, b, 0, 0, SIZE, x);}
    //query(a,b) := [a,b)ã\201«å¯¾ã\201\231ã\202\213æ\234\200å°\217å\200¤ã\202\222æ±\202ã\202\201ã\202\213
    ll query(int a, int b){return query(a, b, 0, 0, SIZE);}
};
*/

/*
http://tenka1-2016-qualb.contest.atcoder.jp/tasks/tenka1_2016_qualB_d
*/


################################################
##### ./data_struture/segtree_lazy_add_sum.h #####
################################################

//(1)å\214ºé\226\223ã\201«ä¸\200æ§\230å\212 ç®\227 (2)å\214ºé\226\223ã\201®å\220\210è¨\210å\200¤
struct segtree{
public:
    const int SIZE = 1 << 18;
    //seg:å\214ºé\226\223ã\201®å\220\210è¨\210å\200¤ lazy:å\214ºé\226\223ã\201«å¯¾ã\201\227ã\201¦ã\200\201å\212 ã\201\210ã\202\213å\200¤ã\201§ã\201¾ã\201 é\201\205å»¶ã\201\227ã\201¦ã\201\204ã\202\213ã\202\202ã\201®
    vector<ll> seg, lazy;//segã\201¯æ¬²ã\201\227ã\201\204æ\203\205å ± lazyã\201¯å\214ºé\226\223ã\201«å¯¾ã\201\231ã\202\213ä¸\200æ§\230ã\201ªå\207¦ç\220\206ã\202\222ç¤ºã\201\231ã\202\202ã\201®
    segtree():seg(SIZE * 2), lazy(SIZE * 2){}
    void lazy_evaluate(int k, int l, int r){//é\201\205å»¶æ\203\205å ±ã\201®é\201©ç\224¨æ\226¹æ³\225
        if(lazy[k] != 0){
            seg[k] += lazy[k];//å\214ºé\226\223[l,r)ã\201«ã\201\231ã\201¹ã\201¦å\220\214ã\201\230å\200¤ã\202\222è¿½å\212 ã\201\231ã\202\213ã\201\223ã\201¨ã\201«ã\201ªã\201£ã\201¦ã\201\204ã\201¦ã\200\201segã\201«ã\201¯å\220\210è¨\210å\200¤ã\201\214å\205¥ã\201£ã\201¦ã\201\204ã\202\213ã\201®ã\201§ã\200\201å\212 ã\201\210ã\202\213å\200¤ã\202\222è¶³ã\201\231
            if(r  - l > 1){
                lazy[k * 2 + 1] += lazy[k];//é\201\205å»¶ã\202\222å•¦ã\201®å-\220ã\201«ä¼\235æ\220¬
                lazy[k * 2 + 2] += lazy[k];//é\201\205å»¶ã\202\222å\217³ã\201®å-\220ã\201«ä¼\235æ\220¬
            }
            lazy[k] = 0;//ã\203\216ã\203¼ã\203\211kã\201¯ä¼\235æ\220¬å®\214äº\206
        }
    }
    void update(int a, int b, int k, int l, int r, int x){
        lazy_evaluate(k, l, r);
        if(r <= a || b <= l) return;
        if(a <= l && r <= b){
            lazy[k] += x; //å\212 ã\201\210ã\202\213
            lazy_evaluate(k, l, r);
        }else{
            update(a, b, k * 2 + 1, l, (l + r) / 2, x);
            update(a, b, k * 2 + 2, (l + r) / 2, r, x);
            seg[k] = seg[k * 2 + 1] + seg[k * 2 + 2]; //å\214ºé\226\223ã\201®å\220\210è¨\210
        }
    }
    ll query(int a, int b, int k, int l, int r){
        lazy_evaluate(k, l, r);
        if(r <= a || b <= l) return 0;//å\220\210è¨\210ã\201«å½±é\237¿ã\201®ã\201ªã\201\204ã\202\202ã\201®
        if(a <= l && r <= b) return seg[k];
        ll x = query(a, b, k * 2 + 1, l, (l + r) / 2);
        ll y = query(a, b, k * 2 + 2, (l + r) / 2, r);
        return x + y; //å•¦å\217³ã\201®å\220\210è¨\210ã\202\222
    }
    //update(a,b,x) := [a,b)ã\202\222å\205¨ã\201¦xã\202\222å\212 ã\201\210ã\202\213
    void update(int a, int b, int x){update(a, b, 0, 0, SIZE, x);}
    //query(a,b) := [a,b)ã\201«å¯¾ã\201\231ã\202\213å\220\210è¨\210å\200¤ã\202\222æ±\202ã\202\201ã\202\213
    ll query(int a, int b){return query(a, b, 0, 0, SIZE);}
};

################################################
######## ./data_struture/segtree_max.hpp #########
################################################

/*
 å\221¼ã\201³ã\201 ã\201\227æ\226¹
 segtree<int> seg(n); // intå\236\213ã\201®è¦\201ç´ ã\201§ nã\201¯è¦\201ç´ ã\201®æ\225°
*/

template <class T> //T : dat[]ã\201®ä¸-è°«ã\201®å\236\213
class segtree{
public:
    int n;
    vector<T> dat;
    segtree(int n_): n(n_){ //n_è¦\201ç´ æ\225°
        n = 1;
        while(n < n_) n *= 2;
        dat.resize(n * 2, 0); //(1) å\210\235æ\234\237å\200¤ã\202\222æ\234\200å°\217ã\201« -INFã\201\213ã\202\202
    }
    void update(int k, T val){ // kç\225ªç\233®ã\201®å\200¤(0-indexed)ã\202\222val ã\201«å¤\211æ\233´
        for (dat[k += n] = val; k > 0; k >>= 1){ // kã\202\222å\220«ã\202\200å\214ºé\226\223ã\201®ã\202¤ã\203\207ã\203\203ã\202¯ã\202¹ã\202\222ä¸\213ã\201\213ã\202\211é \206ã\201«å\210\227æ\214\231
            dat[k>>1] = max(dat[k], dat[k ^ 1]); // (2) å\214ºé\226\223ã\201®æ\234\200å¤§å\200¤ã\201§æ\233´æ\226°
        }
```

```
        }
        T query(int l, int r){ //[l, r)ã\201®å\214ºé\226\223
                T ret = 0; //(3) æ\234\200å¤§å\200¤ã\201«é\226¢ä¿‚ã\202ã\201ªã\201\204å\200¤ -INFã\201\213ã\202\202
                for (l += n, r += n; l < r; l >>= 1, r >>= 1){
                        if(l & 1) ret = max(ret, dat[l++]); //(4) å\214ºé\226\223ã\201®æ\234\200å¤§å\200¤ã\201§æ\233´æ\226°
                        if(r & 1) ret = max(ret, dat[--r]); //(4) å\214ºé\226\223ã\201®æ\234\200å¤§å\200¤ã\201§æ\233´æ\226°
                }
                return ret;
        }
};


/*
segtree ã\201\231ã\201\223ã\201\227é«\230é\200\237å\214\226
æ\233´æ\226°: 1 ç\202¹
ã\202¯ã\202¨ã\203ª: å\214ºé\226\223ã\201®æ\234\200å°\217å\200¤
note: (1)~(4)ã\202\222å¤\211ã\201\210ã\202\213ã\201\223ã\201¨ã\201§å¯¾å¿\234å\217¯è\203½
http://codeforces.com/contest/777/problem/E
(http://codeforces.com/contest/777/submission/25346678)
ã\201\223ã\201®å\225\217é¡\214ã\201§ã\201¯å•®ã\201¯ã\201ªã\201\204

http://codeforces.com/contest/689/problem/D
(http://codeforces.com/contest/689/submission/25354957)
ã\201\223ã\201®å\225\217é¡\214ã\201§ã\201¯segtree_max_slow.hppã\201¨æ¯\224ã\201¹ã\201¦å¤§ã\201\215ã\201ªå•®ã\201\214å\207ºã\201¦ã\201\204ã\202\213
*/


////////////////////////////////////////////////
////// ./data_struture/segtree_max_idx.cpp ///////
////////////////////////////////////////////////

template <class T> //T : dat[]ã\201®ä¸è°«ã\201®å\236\213
class segtree{
public:
        int n;
        T neutral;
        vector<T> dat;
        T func(T l, T r){ //å\214ºé\226\223ã\202\222ã\203\236ã\203¼ã\202¸ã\201\231ã\202\213é\226¢æ\225°
                return max(l, r); /* ã\201\223ã\201\223ã\202\222å¤\211ã\201\210ã\202\213 */
        }
        segtree(int n_, T val): n(n_), neutral(val){ //n_:è¦\201ç´ æ\225° val:å\215\230ä½\215å\205\203
                n = 1;
                while(n < n_) n *= 2;
                dat.resize(n * 2, neutral); //å\210\235æ\234\237å\200¤
        }
        void update(int k, T val){ // kç\225ªç\233®ã\201®å\200¤(0-indexed)ã\202\222val ã\201«å¤\211æ\233´
                for (dat[k += n] = val; k > 0; k >>= 1){ // kã\202\222å\220«ã\202\200å\214ºé\226\223ã\201®ã\202¤ã\203³ã\203\207ã\203\203ã\202¯ã\202¹ã\202\222ä¸\213ã\201\213ã\202\211é \206ã\201«å\210\227æ\214\231
                        dat[k>>1] = func(dat[k], dat[k ^ 1]);
                }
        }
        T query(int l, int r){ //[l, r)ã\201®å\214ºé\226\223
                T ret = neutral;
                for (l += n, r += n; l < r; l >>= 1, r >>= 1){
                        if(l & 1) ret = func(ret, dat[l++]);
                        if(r & 1) ret = func(ret, dat[--r]);
                }
                return ret;
        }
};

int main(void){
        cin >> n >> d >> k;
        rep(i, n) cin >> x[i];
        segtree<pair<ll, int>> seg(n, make_pair(0, 0));
        rep(i, n) seg.update(i, {x[i], -i}); //è¾\236æ\233¸é \206æ\234\200å°\217ã\201®ã\201\237ã\202\201ã\203\236ã\202¤ã\203\212ã\202¹
        return 0;
}

/*
http://yukicoder.me/problems/no/489
(http://yukicoder.me/submissions/156443)
*/


////////////////////////////////////////////////
////// ./data_struture/segtree_max_slow.hpp //////
////////////////////////////////////////////////

/*
 å\221¼ã\201³ã\201 ã\201\227æ\226¹
 segtree<int> seg(n); // intå\236\213ã\201®è¦\201ç´ ã\201§ nã\201¯è¦\201ç´ ã\201®æ\225°
*/

template <class T> //T : dat[]ã\201®ä¸è°«ã\201®å\236\213
class segtree{
public:
        int n;
        vector<T> dat;
        segtree(int n_): n(n_){ //n_è¦\201ç´ æ\225°
                n = 1;
                while(n < n_) n *= 2;
                dat.resize(n * 2, 0); //(1) å\210\235æ\234\237å\200¤ã\202\222æ\234\200å°\217ã\201« -INFã\201\213ã\202\202
        }
        void update(int k, T val){ // kç\225ªç\233®ã\201®å\200¤(0-indexed)ã\202\222val ã\201«å¤\211æ\233´
                k += n - 1; //è\221\211ã\201®ç¯\200ç\202¹
                dat[k] = val;
                while(k > 0){
                        k = (k - 1) / 2;
                        dat[k] = max(dat[k * 2 + 1], dat[k * 2 + 2]); // (2) å\214ºé\226\223ã\201®æ\234\200å¤§å\200¤ã\201§æ\233´æ\226°
                }
        }
        T query(int a, int b, int k, int l, int r){ //[a, b)ã\201®æ\234\200å¤§å\200¤ã\202\222æ±\202ã\202\201ã\202\213
                if(r <= a || b <= l) return 0; //(3) æ\234\200å¤§å\200¤ã\201«é\226¢ä¿‚ã\201ªã\201\204å\200¤ã\201§æ\233´æ\226° -INFã\201\213ã\202\202
                if(a <= l && r <= b) return dat[k];
                else{
                        T vl = query(a, b, k * 2 + 1, l, (l + r) / 2);
                        T vr = query(a, b, k * 2 + 2, (l + r) / 2, r);
                        return max(vl, vr); //(4) å\214ºé\226\223ã\201®æ\234\200å¤§å\200¤ã\201§æ\233´æ\226°
                }
        }
        T query(int a, int b){
                return query(a, b, 0, 0, n);
        }
```

```
};

/*
segtree
æ\233´æ\226°: 1 ç\202¹
ã\202¯ã\202¨ã\203ª: å\214ºé\226\223ã\201®æ\234\200å¤§å\200¤
note: (1)~(4)ã\202\222å¤\211ã\201\210ã\202\213ã\201\223ã\201¨ã\201§å¯¾å¿\234å\217¯è\203½

http://codeforces.com/contest/777/problem/E
(http://codeforces.com/contest/777/submission/25143272)
*/

#####################################################
######## ./data_struture/segtree_min.hpp #########
#####################################################

/*
 å\221¼ã\201³ã\201 ã\201\227æ\226¹
 segtree<int> seg(n); // intå\236\213ã\201®è¦\201ç´ ã\201§ nã\201¯è¦\201ç´ ã\201®æ\225°
*/

template <class T> //T : dat[]ã\201®ä¸ -èº«ã\201®å\236\213
class segtree{
public:
        int n;
        vector<T> dat;
        segtree(int n_): n(n_){ //n_è¦\201ç´ æ\225°
                n = 1;
                while(n < n_) n *= 2;
                dat.resize(n * 2, INF); //(1) å\210\235æ\234\237å\200¤ã\202\222æ\234\200å¤§ã\201«
        }
        void update(int k, T val){ // kç\225ªç\233®ã\201®å\200¤(0-indexed)ã\202\222val ã\201«å¤\211æ\233´
                for (dat[k += n] = val; k > 0; k >>= 1){ // kã\202\222å\220«ã\202\200å\214ºé\226\223ã\201®ã\202¤ã\203³ã\203\207ã\203\203ã\202¯ã\202¹ã\202\222ä¸ \213ã\201\213ã\202\211é \206ã\201«å\210\227æ\214\231
                        dat[k>>1] = min(dat[k], dat[k ^ 1]); // (2) å\214ºé\226\223ã\201®æ\234\200å¤§å\200¤ã\201§æ\233´æ\226°
                }
        }
        T query(int l, int r){ //[l, r)ã\201®å\214ºé\226\223
                T ret = INF; //(3) æ\234\200å°\217å\200¤ã\201«é\226¢ä¿\202ã\201ªã\201\204å\200¤
                for (l += n, r += n; l < r; l >>= 1, r >>= 1){
                        if(l & 1) ret = min(ret, dat[l++]); //(4) å\214ºé\226\223ã\201®æ\234\200å°\217å\200¤ã\201§æ\233´æ\226°
                        if(r & 1) ret = min(ret, dat[--r]); //(4) å\214ºé\226\223ã\201®æ\234\200å°\217å\200¤ã\201§æ\233´æ\226°
                }
                return ret;
        }
};


/*
segtree ã\201\231ã\201\223ã\201\227é«\230é\200\237å\214\226
æ\233´æ\226°: 1 ç\202¹
ã\202¯ã\202¨ã\203ª: å\214ºé\226\223ã\201®æ\234\200å°\217å\200¤
note: (1)~(4)ã\202\222å¤\211ã\201\210ã\202\213ã\201\223ã\201¨ã\201§å¯¾å¿\234å\217¯è\203½
http://arc045.contest.atcoder.jp/tasks/arc045_b
(http://arc045.contest.atcoder.jp/submissions/1151765)

http://codeforces.com/contest/689/problem/D
(http://codeforces.com/contest/689/submission/25354957)
*/

#####################################################
###### ./data_struture/segtree_min_slow.hpp ######
#####################################################

/*
 å\221¼ã\201³ã\201 ã\201\227æ\226¹
 segtree<int> seg(n); // intå\236\213ã\201®è¦\201ç´ ã\201§ nã\201¯è¦\201ç´ ã\201®æ\225°
*/

template <class T> //T : dat[]ã\201®ä¸ -èº«ã\201®å\236\213
class segtree{
public:
        int n;
        vector<T> dat;
        segtree(int n_): n(n_){ //n_è¦\201ç´ æ\225°
                n = 1;
                while(n < n_) n *= 2;
                dat.resize(n * 2, INF); //(1) å\210\235æ\234\237å\200¤ã\202\222æ\234\200å¤§ã\201«
        }
        void update(int k, T val){ // kç\225ªç\233®ã\201®å\200¤(0-indexed)ã\202\222val ã\201«å¤\211æ\233´
                k += n - 1; //è\221\211ã\201®ç¯\200ç\202¹
                dat[k] = val;
                while(k > 0){
                        k = (k - 1) / 2;
                        dat[k] = min(dat[k * 2 + 1], dat[k * 2 + 2]); // (2) å\214ºé\226\223ã\201®æ\234\200å°\217å\200¤ã\201§æ\233´æ\226°
                }
        }
        T query(int a, int b, int k, int l, int r){ //[a, b)ã\201®æ\234\200å¤§å\200¤ã\202\222æ±\202ã\202\201ã\202\213
                if(r <= a || b <= l) return INF; //(3) æ\234\200å°\217å\200¤ã\201«é\226¢ä¿\202ã\201ªã\201\204å\200¤ã\201§æ\233´æ\226°
                if(a <= l && r <= b) return dat[k];
                else{
                        T vl = query(a, b, k * 2 + 1, l, (l + r) / 2);
                        T vr = query(a, b, k * 2 + 2, (l + r) / 2, r);
                        return min(vl, vr); //(4) å\214ºé\226\223ã\201®æ\234\200å°\217å\200¤ã\201§æ\233´æ\226°
                }
        }
        T query(int a, int b){
                return query(a, b, 0, 0, n);
        }
};

/*
segtree
æ\233´æ\226°: 1 ç\202¹
ã\202¯ã\202¨ã\203ª: å\214ºé\226\223ã\201®æ\234\200å°\217å\200¤
note: (1)~(4)ã\202\222å¤\211ã\201\210ã\202\213ã\201\223ã\201¨ã\201§å¯¾å¿\234å\217¯è\203½

http://arc045.contest.atcoder.jp/tasks/arc045_b
(http://arc045.contest.atcoder.jp/submissions/1151282)
*/
```

```
545:
546: ################################################
547: ####### ./data_struture/segtree_monoid.cpp #######
548: ################################################
549:
550: template <class T> //T(ã\203¢ã\203\216ã\202¤ã\203\211) : dat[]ã\201®ä¸-è°«ã\201®å
\236\213
551: class segtree{
552: public:
553:         int n;
554:         T neutral;
555:         vector<T> dat;
556:         T func(T l, T r) { //ä°\214é \205æ¼\224ç®\227å-\220
557:                 int lok, lopen, lclose, rok, ropen, rclose;
558:                 tie(lok, lopen, lclose) = l;
559:                 tie(rok, ropen, rclose) = r;
560:                 int add = min(lopen, rclose);
561:                 return make_tuple(lok + rok + 2 * add, lopen + ropen - add, lclose +
 rclose - add);
562:         }
563:         segtree(int n_, T val): n(n_), neutral(val) { //n_:è¦\201ç´ æ\225° val:å\215
\230ä½\215å\205\203
564:                 n = 1;
565:                 while(n < n_) n *= 2;
566:                 dat.resize(n * 2, neutral); //å\210\235æ\234\237å\200¤
567:         }
568:         void update(int k, T val){ // kç\225ªç\233®ã\201®å\200¤(0-indexed)ã\202\222
val ã\201«å¤\211æ\233´
569:                 k += n - 1;
570:                 dat[k] = val;
571:                 while (k > 0) {
572:                         k = (k - 1) / 2;
573:                         dat[k] = func(dat[k * 2 + 1], dat[k * 2 + 2]);
574:                 }
575:         }
576:         T query(int a, int b, int k, int l, int r) { //[a, b)ã\201®å\214°é\226\223
577:                 if(r <= a || b <= l) return neutral;
578:                 if(a <= l && r <= b) return dat[k];
579:                 else {
580:                         T vl = query(a, b, k * 2 + 1, l, (l + r) / 2);
581:                         T vr = query(a, b, k * 2 + 2, (l + r) / 2, r);
582:                         return func(vl, vr);
583:                 }
584:         }
585:         T query(int a, int b) {
586:                 return query(a, b, 0, 0, n);
587:         }
588: };
589:
590: ################################################
591: ###### ./data_struture/segtree_template.hpp ######
592: ################################################
593:
594: template <class T> //T : dat[]ã\201®ä¸-è°«ã\201®å\236\213
595: class segtree{
596: public:
597:         int n;
598:         T neutral;
599:         vector<T> dat;
600:         T func(T l, T r){ //å\214°é\226\223ã\202\222ã\203\236ã\203¼ã\202¸ã\201\231ã
\202\213é\226¢æ\225°
601:                 return max(l, r); /* ã\201\223ã\201\223ã\202\222å¤\211ã\201\210ã\202
\213 */
602:         }
603:         segtree(int n_, T val): n(n_), neutral(val){ //n_:è¦\201ç´ æ\225° val:å\215
\230ä½\215å\205\203
604:                 n = 1;
605:                 while(n < n_) n *= 2;
606:                 dat.resize(n * 2, neutral); //å\210\235æ\234\237å\200¤
607:         }
608:         void update(int k, T val){ // kç\225ªç\233®ã\201®å\200¤(0-indexed)ã\202\222
val ã\201«å¤\211æ\233´
609:                 for (dat[k += n] = val; k > 0; k >>= 1){ // kã\202\222å\220«ã\202
\200å\214°é\226\223ã\201®ã\202¤ã\203³ã\203\207ã\203\203ã\202¯ã\202¹ã\202\222ä¸ \213ã\201\213ã
\202\211é \206ã\201«å\210\227æ\214\231
610:                         dat[k>>1] = func(dat[k], dat[k ^ 1]);
611:                 }
612:         }
613:         T query(int l, int r){ //[l, r)ã\201®å\214°é\226\223
614:                 /* å\217¯æ\217\233å\207ºã\201ªã\201\204æ\231\202å\215±é\231º */
615:                 T ret = neutral;
616:                 for (l += n, r += n; l < r; l >>= 1, r >>= 1){
617:                         if(l & 1) ret = func(ret, dat[l++]);
618:                         if(r & 1) ret = func(ret, dat[--r]);
619:                 }
620:                 return ret;
621:         }
622: };
623:
624: /*
625: å\221¼ã\201³å\207ºã\201\227æ\226¹
626: ä¾\213
627: segtree<pair<ll, int>> seg(n, make_pair(0, 0));
628:
629: funcã\201®é\203¨å\210\206ã\202\222å¤\211ã\201\210ã\201¦ã\203\236ã\203¼ã\202¸ã\201
\231ã\202\213
630:
631: http://yukicoder.me/problems/no/489
632: (http://yukicoder.me/submissions/156443)
633: */
634:
635: ################################################
636: ###### ./data_struture/SparseTable_max.hpp #######
637: ################################################
638:
639: /*
640:  å\221¼ã\201³ã\201 ã\201\227æ\226¹
641:  int v[1000];
642:  segtree<int> seg(n, v); // intå\236\213ã\201®è¦\201ç´ ã\201§ nã\201¯è¦\201ç´ ã\201®
æ\225°
643: */
644:
645: template <class T> //T : table[][]ã\201®ä¸-è°«ã\201®å\236\213
646: class SparseTable_max{
647: public:
648:         int N, M; //table[N][M]
649:         // table[i][k] := [i, i + 2^k)ã\201®æ\234\200å¤§å\200¤
650:         vector<vector<T>> table;
651:         template<class S> SparseTable_max(int n, S &val): N(n){ // O(nlogn)
652:                 M = 32 - __builtin_clz(N); // M - 1 <= logN < M
653:                 table.resize(N, vector<T>(M));
654:                 for (int i = 0; i < N; ++i){ // [i, i + 1)ã\201¾ã\201§ã\201®å\214°é
\226\223ã\201®æ\234\200å¤§å\200¤
655:                         table[i][0] = val[i];
656:                 }
657:                 for (int k = 0; k < M - 1; ++k){ // [i, i + 2^(k+1))ã\201®å\214°é
\226\223ã\202\222è¨\210ç®\227
658:                         for (int i = 0; i + (1<< k) < N; ++i){
659:                                 // iã\201\213ã\202\2112^(k+1)ã\201®é\225•ã\201\225ã
\201®å\214°é\226\223ã\201®æ\234\200å°\217å\200¤ã\202\2222^kã\201®é\225•ã\201\225ã\201®å\214°
é\226\223ã\201®æ\234\200å¤§å\200¤ã\202\222å\210©ç\224¨ã\201\227ã\201¦æ±\202ã\202\201ã\202
\213
```

```
                                table[i][k + 1] = max(table[i][k], table[i + (1 << k)][k]); // (1)æ\234\200å¤§å\200¤
                        }
                }
        }
        T query(int l, int r){ // O(1) [l, r) ã\201®é\226\223ã\201®æ\234\200å¤§å\200¤
                int k = 31 - __builtin_clz(r - l); //å\214ºé\226\223ã\201®é\225•ã\201\225ã\201®å\215\212å\210\206ä»¥ä¸\212ã\201®å\200¤ ï¼\210k<= r - l < k + 1)
                return max(table[l][k], table[r - (1 << k)][k]); // (2) æ\234\200å¤§å\200¤
        }
};

/*
å\211\215å\207¦ç\220\206(nlogn)
æ\234\200å°\217å\200¤(1)

http://codeforces.com/contest/689/problem/D
(http://codeforces.com/contest/689/submission/25363691)
*/


////////////////////////////////////////////////
////// ./data_struture/SparseTable_min.hpp ///////
////////////////////////////////////////////////

/*
 å\221¼ã\201³ã\201 ã\201\227æ\226¹
 int v[1000];
 segtree<int> seg(n, v); // intå\236\213ã\201®è¦\201ç´ ã\201§ nã\201¯è¦\201ç´ ã\201®æ\225°
*/

template <class T> //T : table[][]ã\201®ä¸-èº«ã\201®å\236\213
class SparseTable{
public:
        int N, M; //table[N][M]
        // table[i][k] := [i, i + 2^k)ã\201®æ\234\200å°\217å\200¤
        vector<vector<T>> table;
        template<class S> SparseTable(int n, S &val): N(n){ // O(nlogn)
                M = 32 - __builtin_clz(N); // M - 1 <= logN < M
                table.resize(N, vector<T>(M));
                for (int i = 0; i < N; ++i){ // [i, i + 1)ã\201¾ã\201§ã\201®å\214ºé\226\223ã\201®æ\234\200å°\217å\200¤
                        table[i][0] = val[i];
                }
                for (int k = 0; k < M - 1; ++k){ // [i, i + 2^(k+1))ã\201®å\214ºé\226\223ã\202\222è¨\210ç®\227
                        for (int i = 0; i + (1<< k) < N; ++i){
                                // iã\201\213ã\202\2112^(k+1)ã\201®é\225•ã\201\225ã\201®å\214ºé\226\223ã\201®æ\234\200å°\217å\200¤ã\202\2222^kã\201®é\225•ã\201\225ã\201®å\214ºé\226\223ã\201®æ\234\200å°\217å\200¤ã\202\222å\210©ç\224¨ã\201\227ã\201¦æ±\202ã\202\201ã\202\213
                                table[i][k + 1] = min(table[i][k], table[i + (1 << k)][k]); // (1)æ\234\200å°\217å\200¤
                        }
                }
        }
        T query(int l, int r){ // O(1) [l, r) ã\201®é\226\223ã\201®æ\234\200å°\217å\200¤
                int k = 31 - __builtin_clz(r - l); //å\214ºé\226\223ã\201®é\225•ã\201\225ã\201®å\215\212å\210\206ä»¥ä¸\212ã\201®å\200¤ ï¼\210k<= r - l < k + 1)
                return min(table[l][k], table[r - (1 << k)][k]); // (2) æ\234\200å°\217å\200¤
        }
};

/*
template <class T> //T : table[][]ã\201®ä¸-èº«ã\201®å\236\213
class SparseTable{
public:
        int N, M; //table[N][M]
        // table[i][k] := [i, i + 2^k)ã\201®æ\234\200å°\217å\200¤
        vector<vector<T>> table;
        template<class S> SparseTable(int n, S &val): N(n){ // O(nlogn)
                M = 32 - __builtin_clz(N); // M - 1 <= logN < M
                table = vector<vector<T>>(N, vector<T>(M, INF)); // (1)æ\234\200å°\217ã\201«é\226¢ä¿\202ã\201ªã\201\204å\200¤ã\201§æ\233´æ\226°
                for (int i = 0; i < N; ++i){ // [i, i + 1)ã\201¾ã\201§ã\201®å\214ºé\226\223ã\201®æ\234\200å°\217å\200¤
                        table[i][0] = val[i];
                }
                for (int k = 1; k < M; ++k){ // [i, i + 2^k)ã\201®å\214ºé\226\223ã\202\222è¨\210ç®\227
                        for (int i = 0; i + (1<< k) <= N; ++i){
                                // iã\201\213ã\202\2112^kã\201®é\225•ã\201\225ã\201®å\214ºé\226\223ã\201®æ\234\200å°\217å\200¤ã\202\2222^(k-1)ã\201®é\225•ã\201\225ã\201®å\214ºé\226\223ã\201®æ\234\200å°\217å\200¤ã\202\222å\210©ç\224¨ã\201\227ã\201¦æ±\202ã\202\201ã\202\213
                                auto first = table[i][k - 1];
                                auto second = table[i + (1<< (k - 1))][k - 1];
                                table[i][k] = (first < second) ? first : second; // (2) å°\217ã\201\225ã\201\204æ\226¹ã\201§æ\233´æ\226°
                        }
                }
        }
        T query(int l, int r){ // O(1) [l, r) ã\201®é\226\223ã\201®æ\234\200å°\217å\200¤
                int k = 31 - __builtin_clz(r - l); //å\214ºé\226\223ã\201®é\225•ã\201\225ã\201®å\215\212å\210\206ä»¥ä¸\212ã\201®å\200¤ ï¼\210k<= r - l < k + 1)
                return min(table[l][k], table[r - (1 << k)][k]); // (3) æ\234\200å°\217å\200¤
        }
};
*/

/*
å\211\215å\207¦ç\220\206(nlogn)
æ\234\200å°\217å\200¤(1)

http://arc045.contest.atcoder.jp/tasks/arc045_b
(http://arc045.contest.atcoder.jp/submissions/1152995)
(http://arc045.contest.atcoder.jp/submissions/1152999)
æ\234\200å\210\235ã\201®å\210\235æ\234\237å\200¤ã\202\222ã\201©ã\201®ã\202\210ã\201\206ã\201«ã\201\231ã\202\213ã\201\213
*/


////////////////////////////////////////////////
////////// ./data_struture/union_find.h ///////
////////////////////////////////////////////////

class DisjointSet{
public:
        vector<int> rank, p;//rank:æ\234¨ã\201®é«\230ã\201\225 p:è¦ªã\201®é \202ç\202²ç\225ªå\217•
        DisjointSet(){}
```

```
        DisjointSet(int size){//é \202ç\202¹ã\201®æ\225°
                rank.resize(size, 0);
                p.resize(size, 0);
                rep(i, size) makeSet(i);
        }
        bool same(int x, int y){ //å\220\214ã\201\230æ\234¨ã\201«ã\201\202ã\202\213ã\201\213
                return findSet(x) == findSet(y);
        }
        void unite(int x, int y){ // æ\234¨ã\201©ã\201\206ã\201\227ã\202\222ã\201\217ã\201£ã\201¤ã\201\221å\202\213
                link(findSet(x), findSet(y));
        }
private:
        void makeSet(int x){
                p[x] = x;
                rank[x] = 0;
        }
        int findSet(int x){ //è¦ªã\202\222æ\216¢ã\201\231ï¼\210ã\203«ã\203¼ã\203\210ã\201¾ã\201§ï¼\211
                if(x != p[x]){
                        p[x] = findSet(p[x]);
                }
                return p[x];
        }
        void link(int x, int y){ //æ\234¨ã\201®é«\230ã\201\225ã\202\222è\200\203æ\205®ã\201\227ã\201¦æ\234¨ã\201©ã\201\206ã\201\227ã\202\222ã\201\217ã\201£ã\201¤ã\201\221ã\202\213
                if(rank[x] > rank[y]){
                        p[y] = x;
                }else{
                        p[x] = y;
                        if(rank[x] == rank[y]) rank[y]++;
                }
        }
};

/*
http://abc049.contest.atcoder.jp/tasks/arc065_b
*/

################################################
################ ./dp/imos2d.cpp #################
################################################

/*
 * äº\214æ¬¡å\205\203ç´¯ç©\215å\222\214(é\225•æ\226¹å½¢å\206\205ã\201®å\200¤ã\201®å\220\210è¨\210ã\202\222å\207ºã\201\231å\225\217é¡\214)
 */

// O(hw)
void sum2D(int h, int w) { // a ã\201®ç¸¦ã\201¨æ¨ªã\201®é\225•ã\201\225
        rep(y, h + 1)rep(x, w + 1) sum[y][x] = 0;
        rep(y, h)rep(x, w) sum[y + 1][x + 1] = a[y][x]; // ã\201¾ã\201\232ã\201¯å\237\213ã\202\201è¾¼ã\202\200
        rep(y, h + 1)rep(x, w) sum[y][x + 1] += sum[y][x]; // æ¨ª
        rep(y, h)rep(x, w + 1) sum[y + 1][x] += sum[y][x]; // ç¸¦
}
// O(1)
int calcSum(int y1, int x1, int y2, int x2) { // æ±\202ã\202\201ã\201\237ã\201\204é\225•æ\226¹å½¢ã\201®å•¦ä¸\212, å\217³ä¸\213ã\201®å°§æ¨\231
        return sum[y2 + 1][x2 + 1] - sum[y2 + 1][x1] - sum[y1][x2 + 1] + sum[y1][x1];
}

/*
"ã\201¾ã\201\232ã\201¯å\237\213ã\202\201è¾¼ã\202\200" ã\201®é\203¨å\210\206ã\201®aã\201¯æ\233¸ã\201\215æ\217\233ã\201\210ã\202\213
aoj 1176

*/

################################################
########## ./dp/Traveling_Salesman.h ############
################################################

const int MAX_N = 20;
vector<pair<int, int> > G[MAX_N]; /* G[u][v] := u->v's weight */
int dp[(1 << MAX_N)][MAX_N];
int pre[(1 << MAX_N)][MAX_N];
int Traveling_Salesman(int n){ /* n := Number of vertices */
        rep(i, (1 << n))rep(j, n)dp[i][j] = INF; /* Initialization */
        dp[0][0] = 0;
        for (int mask = 0; mask < (1 << n); ++mask){
                for (int u = 0; u < n; ++u){ /* Current vertex */
                        for(auto p : G[u]){ /* Next vertex */
                                int v = p.first, w = p.second;
                                if((mask & (1 << v)) == 0){
                                        if(dp[mask | (1 << v)][v] > dp[mask][u] + w){
                                                dp[mask | (1 << v)][v] = dp[mask][u] + w;
                                                pre[mask | (1 << v)][v] = u;
                                        }
                                }
                        }
                }
        }
        return dp[(1 << n) - 1][0];
}
vector<int> Restoration(int n){
        vector<int> root;
        int mask = (1 << n) - 1, v = 0;
        root.push_back(v);
        while(mask != 0){
                int u = pre[mask][v]; /* u -> v */
                root.push_back(u);
                mask ^= (1 << v);
                v = u;
        }
        reverse(root.begin(), root.end());
        return root;
}

/*
http://judge.u-aizu.ac.jp/onlinejudge/description.jsp?id=DPL_2_A
çµ\214è•¯å¾©å\205\203\201¯æ¤\234è¨¼ã\201\227ã\201¦ã\201\204ã\201ªã\201\204

*/

################################################
############ ./geometry/template.cpp #############
################################################

class Point { // ç\202¹
public:
    double x, y;
    Point(double x = 0, double y = 0):x(x), y(y){}
    Point operator + (Point p) const { return Point(x + p.x, y + p.y); }
    Point operator - (Point p) const { return Point(x - p.x, y - p.y); }
    Point operator * (double a) const { return Point(a * x, a * y); }
    Point operator / (double a) const { return Point(x / a, y / a); }
    double abs() const { return sqrt(norm()); }
```

```
    double norm() const { return x * x + y * y; }
    // bool operator < (const Point &p) const { return x != p.x ? x < p.x : y < p.y;
}
    bool operator < (const Point &p) const { // è³ªå•®ã\202\222è¨±å®¹ã\201\227ã\201¦æ¯\224è¼\203
        return x + EPS < p.x || (eq<double>(x, p.x) && y + EPS < p.y);
    }
    bool operator == (const Point &p) const { return (eq<double>(x, p.x) && eq<double>(y, p.y)); }
};
using Vector = Point;
using Polygon = vector<Point>; // å¤\232è§\222å½¢

double dot(const Vector& a, const Vector& b) { return a.x * b.x + a.y * b.y; } // ã\203\231ã\202¯ã\203\210ã\203«aã\201¨bã\201®å\206\205ç©\215
double cross(const Vector& a, const Vector& b) { return a.x * b.y - a.y * b.x; } // ã\203\231ã\202¯ã\203\210ã\203«aã\201¨bã\201®å¤\226ç©\215
double length2(const Point& a) { return a.norm(); } // é\200\232å¸¸ã\201®é\225·ã\201\225ã\201®2ä¹\227
double length(const Point& a) { return a.abs(); } // é\200\232å¸¸ã\201®é\225·ã\201\225
Point rotationalTransfer(Point c, double r, double deg) { // cã\202\222ä¸-å¿\203ã\201¨ã\201\227ã\201¦å\215\212å¾\204rã\201®å\206\206å\221¨ä¸\212ã\201®degå°¦ã\201®ä½\215ç½®å°§æ¨\231
    double rad = PI * deg / 180.0; return c + Point(cos(rad), sin(rad)) * r;
}
// (x, y, z) ã\201®ç\202¹ã\202\222å\205\211æ°\220(xyå°§æ¨\231ã\201§ã\201®è§\222å°¦ã\201\214thetaå°¦, xyå¹³é\235¢ã\201\213ã\202\211zæ\226¹å\220\221ã\201¸ã\201®è§\222å°¦ã\201\214phiå°¦ã\201®æ\231\202ã\201®)ã\201\213ã\202\211ã\201¦ã\202\211ã\201\227ã\201\237æ\231\202ã\201®å½±ã\201®xyå°§æ¨\231
Point Shadow(double x, double y, double z, double theta, double phi) {
    theta = PI * theta / 180.0, phi = PI * phi / 180.0;
    return Point(x - z / tan(phi) * cos(theta), y - z / tan(phi) * sin(theta));
}

enum ccw_t {
    COUNTER_CLOCKWISE = 1, // p0->p1 å\217\215æ\231\202è¨\210å\233\236ã\202\212ã\201®æ\226¹å\220\221ã\201«p2
    CLOCKWISE = -1, // p0->p1 æ\231\202è¨\210å\233\236ã\202\212ã\201®æ\226¹å\220\221ã\201«p2
    ONLINE_BACK = 2, // p2->p0->p1 ã\201®é \206ã\201§ç\233´ç•\232ä¸\212ã\201§p2
    ONLINE_FRONT = -2, // p0->p1->p2 ã\201®é \206ã\201§ç\233´ç•\232ä¸\212p2
    ON_SEGMENT = 0, // p0->p2->p1 ã\201®é \206ã\201§ç•\232å\210\206p0p1ä¸\212ã\201«p2
};
ccw_t ccw(Point p0, Point p1, Point p2) {
    Vector a = p1 - p0, b = p2 - p0;
    if ( cross(a, b) > EPS  )  return COUNTER_CLOCKWISE;
    if ( cross(a, b) < -EPS )  return CLOCKWISE;
    if ( dot(a, b) < -EPS )    return ONLINE_BACK;
    if ( a.norm() < b.norm() ) return ONLINE_FRONT;
    return ON_SEGMENT;
}

class Segment { //ç•\232å\210\206
public:
    Point p1, p2;
    Segment(){}
    Segment(Point p1, Point p2):p1(p1), p2(p2){}
};
using Line = Segment;

// *** å¤\232è§\222å½¢ ***
// IN := 2, ON := 1, OUT := 0
vector<Segment> getPolygonSegument(const Polygon& p) { //å¤\232è§\222å½¢ã\201®ç\202¹ã\201\213ã\202\211å¤\232è§\222å½¢ã\201®è¾¹ã\202\222æ±\202ã\202\201ã\202\213
    vector<Segment> ret;
    rep(i, p.size() - 1) ret.push_back(Segment(p[i], p[i + 1]));
    ret.push_back(Segment(p[p.size() - 1], p[0]));
    return ret;
}
int contains(const Polygon& g, const Point& p){ // å¤\232è§\222å½¢gã\201®ä¸-ã\201«ç\202¹pã\201\214å\220«ã\201¾ã\202\214ã\201¦ã\201\204ã\202\213ã\201\213
    int n = g.size(); bool x = false;
    for (int i = 0; i < n; ++i) {
        Point a = g[i] - p, b = g[(i + 1) % n] - p;
        if ( abs(cross(a, b)) < EPS && dot(a, b) < EPS ) return 1;
        if (a.y > b.y) swap(a, b);
        if (a.y < EPS && EPS < b.y && cross(a, b) > EPS ) x = !x;
    }
    return (x ? 2 : 0);
}
Polygon andrewScan(Polygon s) { // å\207¸å\214\205(æ\234\200ã\202\202å•¦ã\201«ã\201\202ã\202\213é \202ç\202¹ã\201\213ã\202\211)
    Polygon u, l;
    if (s.size() < 3) return s;
    sort(s.begin(), s.end()); // x, yã\202\222å\237ºæº\226ã\201«æ\230\207é \206ã\202½ã\203¼ã\203\210
    // xã\201\214å°\217ã\201\225ã\201\204ã\202\202ã\201®ã\201\213ã\202\2112ã\201¤ uã\201«è¿½å\212
    u.push_back(s[0]), u.push_back(s[1]);
    // x ã\201\214å¤§ã\201\215ã\201\204ã\202\202ã\201®ã\201\213ã\202\2112ã\201¤1ã\201«è¿½å\212
    l.push_back(s[s.size() - 1]), l.push_back(s[s.size() - 2]);
    // å\207¸å\214\205ã\201®ä¸\212é\203¨ã\202\222ç\224\237æ\210\220
    for (int i = 2; i < s.size(); i++) {
        for (int n = u.size(); n >= 2 && ccw(u[n - 2], u[n - 1], s[i]) != CLOCKWISE; n--){
            u.pop_back();
        }
        u.push_back(s[i]);
    }
    // å\207¸å\214\205ã\201®ä¸\213é\203¨ã\202\222ç\224\237æ\210\220
    for (int i = s.size() - 3; i >= 0; i--) {
        for (int n = l.size(); n >= 2 && ccw(l[n - 2], l[n - 1], s[i]) != CLOCKWISE; n--){
            l.pop_back();
        }
        l.push_back(s[i]);
    }
    // æ\231\202è¨\210å\233\236ã\202\212ã\201«ã\201ªã\202\213ã\202\210ã\201\206ã\201«å\207¸å\214\205ã\201®ç\202¹ã\201®å\210\227ã\202\222ç\224\237æ\210\220
    reverse(l.begin(), l.end());
    for (int i = u.size() - 2; i >= 1; i--) l.push_back(u[i]);
    return l;
}


// *** ç•\232å\210\206ã\201®äº¤å•®å\210¤å®\232 ***
bool intersect(const Point& p1, const Point& p2, const Point& p3, const Point& p4) {
    return ( ccw(p1, p2, p3) * ccw(p1, p2, p4) <= 0 &&
             ccw(p3, p4, p1) * ccw(p3, p4, p2) <= 0 );
}
bool intersect(const Segment& s1, const Segment& s2) { // äº¤å•®ã\201\227ã\201¦ã\201\204ã\201\237ã\202\211true
    return intersect(s1.p1, s1.p2, s2.p1, s2.p2);
}
//*** ç•\232å\210\206ã\201®äº¤ç\202¹ ***
Point getCrossPoint(Segment s1, Segment s2) { // ç•\232å\210\206ã\201®äº¤ç\202¹ã\201\214å-\230å\234¨ã\201\231ã\202\213ã\201\213ã\202\211è³¿ã\201¹ã\201\237å¾\214ã\201¤ã\201\213ã\201\206ã\201\223ã\201¨
    Vector base = s2.p2 - s2.p1;
```

```
    double d1 = abs(cross(base, s1.p1 - s2.p1)), d2 = abs(cross(base, s1.p2 - s2.p1)
);
    double t  = d1 / (d1 + d2);
    return s1.p1 + (s1.p2 - s1.p1) * t;
}
// *** è•\235é\233¢ ***
double getDistance(Point& a, Point& b) { // ç\202¹aã\201¨ç\202¹bã\201®è•\235é\233¢
    return length(a - b);
}
double getDistanceLP(Line& l, Point& p) { // ç\233´ç•\232sã\201¨ç\202¹pã\201®è•\235é
\233¢
    return length(cross(l.p2 - l.p1, p - l.p1) / length(l.p2 - l.p1));
}
double getDistanceSP(Segment s, Point p) { // ç•\232å\210\206sã\201¨ç\202¹pã\201®è•
\235é\233¢
    if( dot(s.p2 - s.p1, p - s.p1) < EPS ) return length(p - s.p1);
    if( dot(s.p1 - s.p2, p - s.p2) < EPS ) return length(p - s.p2);
    return getDistanceLP(s, p);
}
double getDistanceSS(const Segment& s1, const Segment& s2) { // ç•\232å\210\206s1ã
\201¨ç•\232å\210\206s2ã\201®äº¤ç\202¹
    if( intersect(s1, s2) ) return 0.0; //äº¤ã\202\217ã\201£ã\201¦ã\201\204ã\202\213
ã\201¨ã\201\215
    return min(min(getDistanceSP(s1, s2.p1), getDistanceSP(s1, s2.p2)),
               min(getDistanceSP(s2, s1.p1), getDistanceSP(s2, s1.p2)));
}
double getDistancePolP(const Polygon& pol, const Point& p) { // å¤\232è§\222å½¢polã
\201¨ç\202¹pã\201®è•\235é\233¢
    if(contains(pol, p) != 0) return 0.0; // ç\202¹ã\201\214å¤\232è§\222å½¢ã\201®å
\206\205é\203¨ã\201«å\220«ã\201¾ã\202\214ã\201¦ã\201\204ã\202\213
    double ret = 1e9;
    for(Segment& u : getPolygonSegument(pol)) ret = min(ret, getDistanceSP(u, p));
    return ret;
}
double getDistancePolPol(const Polygon& p1, const Polygon& p2) { // å¤\232è§\222å½¢p
1ã\201¨p2ã\201®è•\235é\233¢
    for(const Point& p : p1) if(contains(p2, p) != 0) return 0.0; // p1ã\201®ç\202¹ã
\201\214å¤\232è§\222å½¢p2ã\201®ä¸ã\201«å\220«ã\201¾ã\202\214ã\201¦ã\201\204ã\202\213
    for(const Point& p : p2) if(contains(p1, p) != 0) return 0.0; // p2ã\201®ç\202¹ã
\201\214å¤\232è§\222å½¢p1ã\201®ä¸ã\201«å\220«ã\201¾ã\202\214ã\201¦ã\201\204ã\202\213
    double ret = 1e9;
    for(const Segment& u : getPolygonSegument(p1))for(const Segment& v : getPolygonS
egument(p2)) {
        ret = min(ret, getDistanceSS(u, v));
    }
    return ret;
}


class Rectangle { // é\225•æ\226¹å½¢
public:
    // 3 2
    // 0 1 (å\217\215æ\231\202è¨\210å\233\236ã\202\212ã\201«é\225•æ\226¹å½¢ã\201®é
\202ç\202¹ã\202\222ã\201\204ã\202\214ã\202\213ã\201\223ã\201¨)
    vector<Point> p; // ç\202¹ã\202\222é \206ç\225ªã\201«ã\201\204ã\202\214ã\202\213
ã\201\223ã\201¨
    Rectangle(vector<Point>&p):p(p) {
        rep(i, 3) reps(j, i + 1, 4) { //é\201©å½\223ã\201ªé \206ç\225ªã\201«ã\201
\204ã\202\214ã\201¦ã\202\202å¤§ã¸ \210å¤«ã\201ªã\202\210ã\201\206ã\201«?
            int cnt = 0;
            rep(k, 4) if(k != i && k != j) {
                cnt += ccw(p[i], p[j], p[k]) == COUNTER_CLOCKWISE;
            }
            if(cnt == 2) {
                swap(p[i + 1], p[j]);
                break;
            }
        }
    }
    bool intersect(const Segment& s) { // ç•\232å\210\206sã\201¨é\225•æ\226¹å½¢ã\201
®å°\221ã\201ªã\201\217ã\201¨ã\202\2021è¾ºã\201\214äº¤å•®ã\201\227ã\201¦ã\201\204ã\202\214ã
\201°true
        bool flag = false;
        rep(i, 4) flag |= ::intersect(s, Segment(p[i], p[(i + 1) % 4]));
        return flag;
    }
    bool contain(const Point& pp) { // ç\202¹ppã\201\214é\225•æ\226¹å½¢å\206\205ã
\201«å\220«ã\201¾ã\202\214ã\202\214ã\201°(è¾ºã\202\222å\220«ã\201¾ã\201ªã\201\204)true
        bool flag = true;
        rep(i, 4) flag &= ccw(p[i], p[(i + 1) % 4], pp) == COUNTER_CLOCKWISE;
        return flag;
    }
    bool contain(const Segment& s) { // ç•\232å\210\206sã\201\214é\225•æ\226¹å½¢å
\206\205ã\201«å\220«ã\201¾ã\202\214ã\202\214ã\201°(è¾ºã\202\222å\220«ã\201¾ã\201ªã\201\204)t
rue
        return contain(s.p1) && contain(s.p2);
    }
};


class Circle {
public:
        Point c;
        double r;
        Circle(Point c = Point(), double r = 0.0):c(c), r(r) {}
}
double arg(Vector p) { return atan2(p.y, p.x); }
Vector polar(double a, double b) { return Point(cos(r) * a, sin(r) * a); }
pair<Point, Point> getCrossPoints(Circle c1, Circle c2) {
        assert(intersect(c1, c2));
        double b = abs(c1.c - c2.c);
        double a = acos(c1.r * c1.r + d * d - c2.r * c2.r) / (2 * c1.r * d);
        double t = arg(c2.c - c1.c);
        return make_pair(c1.c + polar(c1.r, t + a), c1.c + polar(c1.r, t - a));
}

/*
http://judge.u-aizu.ac.jp/onlinejudge/description.jsp?id=2009
http://judge.u-aizu.ac.jp/onlinejudge/description.jsp?id=1157&lang=jp
http://judge.u-aizu.ac.jp/onlinejudge/description.jsp?id=2402
*/

##################################################
########## ./geometry/types_triangles.h ##########
##################################################

class Types_triangles{
public:
        int n;
        vector<int> y, x;//yåº§æ¨\231ã\200\200xåº§æ¨\231
        const double EPS = 1e-10;
        long long cnt_chokaku = 0, cnt_donkaku = 0, cnt_eikaku = 0;
        Types_triangles(const vector<int> &ty, const vector<int> &tx, int size):y(ty
), x(tx), n(size){//n:=åº§æ¨\231ã\201®æ\225°
                count();
        }
private:
        void count(){
                for (int i = 0; i < n; ++i){
                        //-M_PI ~ M_PI(-180 ~ 180)
                        vector<double> angle;//(x[i], y[i])ã\202\222å\216\237ç\202¹ã
\201¨ã\201\227ã\201\237å\201\217è§\222
```

```
                        for (int j = 0; j < n; ++j){
                                if(j == i) continue;
                                angle.push_back(atan2(y[j] - y[i], x[j] - x[i]));
                        }
                        sort(angle.begin(), angle.end());
                        for (int j = 0; j < n - 1; ++j){//2å\221¨ç\233®ã\202\222ä½
\234ã\202\213
                                angle.push_back(angle[j] + M_PI*2);
                        }
                        for (int j = 0; j < n - 1; ++j){
                                cnt_chokaku += upper_bound(angle.begin(), angle.end(
), angle[j] + M_PI/2.0 + EPS) - lower_bound(angle.begin(), angle.end(), angle[j] + M_PI/2.0
 - EPS);
                                cnt_donkaku += lower_bound(angle.begin(), angle.end(
), angle[j] + M_PI) - upper_bound(angle.begin(), angle.end(), angle[j] + M_PI/2.0 + EPS);
                        }
                }
                cnt_eikaku = (ll)n * (n - 1) * (n - 2) / 6 - cnt_chokaku - cnt_donka
ku;
        }
};

/*
å°§æ¨\231ã\201\214ã\201\202ã\201\237ã\201\210ã\202\211ã\202\214ã\201¦ã\200\201ã\201
\235ã\201®ä¸ -ã\201§ã\201§ã\201\215ã\202\213ä¸ \211è§\222å½¢ã\201®ã\201ªã\201\213ã\201§ã\200
\201
é\213-è§\222ã\200\201é\210\215è§\222ã\200\201ç\233´è§\222ã\201\214ã\201\235ã\202\214
ã\201\236ã\202\214ã\201\204ã\201\217ã\201¤ã\201\232ã\201¤ã\201§ã\201\215ã\202\213ã\201\213ã
\202\222æ±\202ã\202\201ã\202\213ã\200\202
http://abc033.contest.atcoder.jp/tasks/abc033_d
*/

////////////////////////////////////////////////
/////////////// ./graph/dijkstra.h ///////////////
////////////////////////////////////////////////

const int MAX_N = 210;
using TYPE = double; // è•\235é\233¢ã\201®å\236\213ã\202\222å\205¥ã\202\214ã\202\213
vector<pair<int, TYPE> > G[MAX_N];
vector<TYPE> dijkstra(int start){
        vector<TYPE> dist(MAX_N, INFF);
        dist[start] = 0;//dist[i] :=ã\200\200start->iã\201¾ã\201§ã\201®æ\234\200ç
\237-è•\235é\233¢
        priority_queue<pair<TYPE, int>, vector<pair<TYPE, int> >, greater<pair<TYPE,
 int> > >  que;
        que.push(make_pair(0, start));
        while(!que.empty()){
                TYPE cost; int u;//ä»\212ã\201¾ã\201§ã\201«ã\201\213ã\201\213ã\201£ã
\201\237æ\231\202é\226\223 ç\217¾å\234¨ã\201®é \202ç\202¹
                cost = que.top().first, u = que.top().second;
                que.pop();
                if(dist[u] < cost) continue;
                for (auto tmp : G[u]){
                        int v = tmp.first; TYPE time = tmp.second;//é\232£æ\216¥ã
\201\231ã\202\213é \202ç\202¹ ã\201\235ã\201®é \202ç\202¹ã\201¾ã\201§è¡\214ã\201\217æ\231
\202é\226\223
                        if(dist[v] > dist[u] + time){//u->v
                                dist[v] = dist[u] + time;
                                que.push(make_pair(dist[v], v));
                        }
                }
        }
        return dist;
}


/*
http://joi2016yo.contest.atcoder.jp/tasks/joi2016yo_e
(http://joi2016yo.contest.atcoder.jp/submissions/1201083)
http://judge.u-aizu.ac.jp/onlinejudge/description.jsp?id=2402
*/


////////////////////////////////////////////////
/////// ./graph/dijkstra_.Restoration.cpp ////////
////////////////////////////////////////////////

const int MAX_N = 210;
vector<pair<int, int> > G[MAX_N];
vector<int> dijkstra(int start, int goal){//ã\202¹ã\202¿ã\203¼ã\203\210ã\201¨ã\202´ã
\203¼ã\203«ã\202\222é\200\206ã\201«
        vector<long long> dist(MAX_N, INF);
        vector<int> pre(MAX_N, -1);//pre[i] := iã\201®å\211\215ã\201®é \202ç\202¹
    dist[start] = 0;//dist[i] :=ã\200\200start->iã\201¾ã\201§ã\201®æ\234\200ç\237-è•
\235é\233¢
    priority_queue<pair<int, int>, vector<pair<int, int> >, greater<pair<int, int> >
> que;
    que.push(make_pair(0, start));
    while(!que.empty()){
        int cost, u, t;//ä»\212ã\201¾ã\201§ã\201«ã\201\213ã\201\213ã\201£ã\201\237æ
\231\202é\226\223 ç\217¾å\234¨ã\201®é \202ç\202¹
        cost = que.top().first, u = que.top().second;
        que.pop();
        if(dist[u] < cost) continue;
        for (auto tmp : G[u]){
            int v = tmp.first, time = tmp.second;//é\232£æ\216¥ã\201\231ã\202\213é
\202ç\202¹ ã\201\235ã\201®é \202ç\202¹ã\201¾ã\201§è¡\214ã\201\217æ\231\202é\226\223
            if(dist[v] > dist[u] + time){//u->v
                dist[v] = dist[u] + time;
                pre[v] = u;
                que.push(make_pair(dist[v], v));
            }else if(dist[v] == dist[u] + time){//è¾\236æ\233¸é \206æ\234\200å°\217
                pre[v] = min(pre[v], u);
            }
        }
    }
    //çµ\214è•¯å¾©å\205\203
    vector<int> path;
    int s = start, t = goal;
        for (; t != s; t = pre[t]) path.push_back(t);
        path.push_back(s);
        return path; //start-> * -> goal startã\201\213ã\202\211goalã\201¾ã\201§ã
\201®æ\234\200ç\237-çµ\214è•¯(è¾\236æ\233¸é \206æ\234\200å°\217)
}

/*
http://yukicoder.me/problems/no/160
*/

////////////////////////////////////////////////
////////////// ./graph/dijkstra_ex.h //////////////
////////////////////////////////////////////////

int dist[110][30]; // tyouten time := cost
using tup = tuple<int, int, int>;
int dijkstra(int start, int goal, int limit){
    dist[start][0] = 0;
    rep(i, 110)rep(j, 30) dist[i][j] = INF;
    priority_queue<tup, vector<tup>, greater<tup>> que; // cost v time
    que.push(make_tuple(0, start, 0));
    while(!que.empty()) {
        int cost, u, time; tie(cost, u, time) = que.top(); que.pop();
```

```
        //cout << "cost u time " << cost << " " << u << " " << time << endl;
        if(u == goal) return cost;
        if(dist[u][time] < cost) continue;
        for(auto tmp : G[u]) {
            int v, nc, nt; tie(v, nc, nt) = tmp;
            int ncost = cost + nc;
            int ntime = time + nt;
            if(ntime > limit) continue;
            if(dist[v][ntime] > ncost) {
                dist[v][ntime] = ncost;
                que.push(make_tuple(ncost, v, ntime));
            }
        }
    }
    return INF; // -1
}

/*
aoj 2585
*/

################################################
############## ./graph/euler_tour.h ##############
################################################

class Euler_Tour{
public:
        vector<vector<int> > g;
        //begin[v],end[v]ã\201¯ã\201\235ã\202\214ã\201\236ã\202\214vã\201\214ã\202ªã\202¤ã\203©ã\203¼ã\203\204ã\202¢ã\203¼ä¸\212ã\201§æ\234\200å\210\235ã\201¨æ\234\200å¾\214ã\201«ç\217¾ã\202\214ã\202\213ã\202¤ã\203³ã\203\207ã\203\203ã\202¯ã\202¹
        //[begin[v], end[v])ã\201\214vã\202\222æ ¹ã\201¨ã\201\231ã\202\213é\203¨å\210\206æ\234¨
        vector<int> euler_tour, begin, end;
        Euler_Tour(int n) : g(n), begin(2 * n), end(2 * n){};//nã\201¯é \202ç\202¹æ\225°
        int k = 0, root = 0;
        void dfs(int curr, int par){//ç\217¾å\234¨ã\200\200parent
                begin[curr] = k;
                euler_tour.push_back(curr);
                k++;
                for(auto next : g[curr]){
                        if(next == par) continue;
                        dfs(next, curr);
                        euler_tour.push_back(curr);
                        k++;
                }
                end[curr] = k;
        }
};

/*
http://codeforces.com/contest/383/problem/C
*/

################################################
################ ./graph/flow.hpp ################
################################################

const int MAX_V = 2010; //å¿\205è¦\201ã\201ªé \202ç\202¹æ\225°
struct Flow{
        struct edge{
                int to, cap, rev;
        };
        vector<edge> G[MAX_V];//é\232£æ\216¥ã\203ªã\202¹ã\203\210
        bool used[MAX_V];

        void add_edge(int from, int to, int cap){
                G[from].push_back((edge){to, cap, (int)G[to].size()});//from -> to
                G[to].push_back((edge){from, 0, (int)G[from].size() - 1});//to -> from
        }
        //å¢\227å\212 ã\203\221ã\202¹ã\202\222æ\216¢ã\201\231
        int dfs(int v, int t, int f){
                if(v == t) return f;
                used[v] = true;
                for (int i = 0; i < G[v].size(); ++i){
                        edge &e = G[v][i];
                        if(!used[e.to] && e.cap > 0){
                                int d = dfs(e.to, t, min(f, e.cap));
                                if(d > 0){
                                        e.cap -= d;
                                        G[e.to][e.rev].cap += d;
                                        return d;
                                }
                        }
                }
                return 0;
        }
        //sã\201\213ã\202\211tã\201¸ã\201®æ\234\200å¤§æµ\201
        int max_flow(int s, int t){
                int flow = 0;
                while(1){
                        memset(used, 0, sizeof(used));
                        int f = dfs(s, t, INF);
                        if(f == 0) return flow;
                        flow += f;
                }
        }
};


/*
æ\234\200å¤§æµ\201 2é\203¨ã\203\236ã\203\203ã\203\201ã\203³ã\202°
http://codeforces.com/contest/777/problem/B
*/

################################################
################ ./graph/lca.cpp #################
################################################

class Tree_lca{
public:
        static const int MAXLOG_V = 25;
        vector<vector<int> > G;
        int V, root;//é \202ç\202¹æ\225° æ ¹
        // parent[k][v] := é \202ç\202¹vã\201\213ã\202\2112^kå\233\236è¦ªã\201´ã\201«ç\231»ã\201£ã\201¦å\210°é\201\224ã\201\231ã\202\213é \202ç\202¹ (é\200\232ã\202\212é\201\216ã\201\216ã\202\213ã\201¨-1)
        vector<int> parent[MAXLOG_V];
        vector<int> depth;//æ ¹ã\201\213ã\202\211ã\201®æ•±ã\201\225

        Tree_lca(int V, int root) : V(V), root(root){
                G.resize(V);
                for (int i = 0; i < MAXLOG_V; ++i) parent[i].resize(V);
                depth.resize(V);
        }
        void unite(int u, int v){ //u-vã\202\222unite
                G[u].push_back(v);
                G[v].push_back(u);
        }
        int dist(int u, int v){ // u-v ã\201®è•\235é\233¢
```

```
                int p = lca(u, v);
                return (depth[u] - depth[p]) + (depth[v] - depth[p]);
        }
        void init(){ //parent[0]ã\201¨depthã\202\222å\210\235æ\234\237å\214\226
                dfs(root, -1, 0);
                //parentã\202\222å\210\235æ\234\237å\214\226 2^kã\201\224ã\201¨ã\201®ã\203\206ã\203¼ã\203\226ã\203«ã\202\222ä½\234æ\210\220
                for (int k = 0; k + 1 < MAXLOG_V; ++k){
                        for (int v = 0; v < V; ++v){
                                if(parent[k][v] < 0) parent[k + 1][v] = -1;//root(rotã\202\210ã\202\212ä¸\212)ã\201\214è¦ª
                                else parent[k + 1][v] = parent[k][parent[k][v]];//ã\203\200ã\203\226ã\203ªã\203³ã\202ºã\201§2å\200\215ä¸\212ã\201®è¦ªã\202\222æ±\202ã\202\201ã\202\213
                        }
                }
        }
private:
        //i¼\221ã\201¤ä¸\212ã\201®è¦ªã\201¨æ•±ã\201\225ã\202\222è¨-å®\232
        void dfs(int v, int p, int d){//ã\203\216ã\203¼ã\203\211ç\225ªå\217•ã\200\200è¦ªã\201®ã\203\216ã\203¼ã\203\211ç\225ªå\217•ã\200\200æ•±ã\201\225
                parent[0][v] = p;
                depth[v] = d;
                for(auto u : G[v]){
                        if(u != p) dfs(u, v, d + 1);
                }
        }
        //u, vã\201®LCAã\202\222æ±\202ã\202\201ã\202\213
        int lca(int u, int v){
                if (depth[u] > depth[v]) swap(u, v);
                for (int k = 0; k + 1 < MAXLOG_V; k++){ //uã\201\214vã\201¨å\220\214ã\201\230é«\230ã\201\225ã\201«ã\201ªã\202\213ã\201¾ã\201§è¦ªã\202\222è¾¿ã\202\213
                        if (((depth[v] - depth[u]) >> k) & 1) {
                                v = parent[k][v];//bitã\201\214ç«\213ã\201£ã\201¦ã\201\204ã\202\214ã\201°ã\200\201è¦ªã\202\2222^kè¾¿ã\201£ã\201¦ã\201\204ã\201\217
                        }
                }
                if (u == v) return u;
                //2å\210\206æ\216¢ç´¢ã\201§LCAã\202\222æ±\202ã\202\201ã\202\213ã\200\200LCAã\202\210ã\202\212ä¸\212ã\201®è¦ªï¼(LA)ã\201¯å¸¸ã\201«å\220\214ã\201\230ã\201«ã\201ªã\202\213
                for (int k = MAXLOG_V - 1; k >= 0; k--) {
                        if (parent[k][u] != parent[k][v]){//LCAã\201«ã\201\237ã\201©ã\202\212ç\235\200ã\201\217é\231\220ç\225\214ã\201¾ã\201§ã\200\201ä¸\212ã\202\222ç\233®æ\214\207ã\201\233ã\201°ã\201\204ã\201\204
                                u = parent[k][u];
                                v = parent[k][v];
                        }
                }
                return parent[0][u];//1ã\201¤ä¸\212ã\201\214LCA
        }
};

/*
init()ã\202\222å¿\230ã\202\214ã\201ªã\201\204ã\202\210ã\201\206ã\201«
*/

////////////////////////////////////////////////
////////////////// ./graph/scc.h ///////////////////
////////////////////////////////////////////////

///å¼•é\200£çµ\220æ\210\220å\210\206å\210\206è§£ (Kosaraju)///
#define MAX_V 10000//é \202ç\202¹æ\225°

int V;
vector<int> G[MAX_V], rG[MAX_V];
vector<int> vs;
bool used[MAX_V];
//cmp[v] = cmp[U]ã\201ªã\202\211ã\200\201é \202ç\202¹u, vã\201¯å\220\214ã\201\230å¼•é\200£çµ\220æ\210\220å\210\206
//cmpã\201«å\205¥ã\201£ã\201¦ã\201\204ã\202\213ç\225ªå\217•ã\201¯é\200£çµ\220ã\201\227ã\201¦ã\201\204ã\202\213ã\202ºã\203©ã\203\225ã\201\224ã\201¨ã\201«ã\200\201ã\203\210ã\203\235ã\203-ã\202¸ã\202«ã\203«ã\202½ã\203¼ã\203\210ã\201\225ã\202\214ã\201¦ã\201\204ã\201¦ã\200\201ã\203«ã\203¼ã\203\227é\203¨å\210\206ã\201¯å\220\214ã\201\230ç\225ªå\217•ã\201«ã\201ªã\201£ã\201¦ã\201\204ã\202\213ã\200\202
int cmp[MAX_V];//cmp[v] := é \202ç\202¹vã\201\214å\220«ã\201¾ã\202\214ã\202\213é\200£çµ\220æ\210\220å\210\206ã\201\214ã\201©ã\202\214ã\201ªã\201®ã\201\213ã\202\222ç¤ºã\201\231ç\225ªå\217•
//é\232£æ\216¥ã\203ªã\202¹ã\203\210ã\202\222ä½\234ã\202\213
void add_edge(int from, int to){//0origin
        G[from].push_back(to);//ä¸\216ã\201\210ã\202\211ã\202\214ã\201\237æ\234\211å\220\221ã\202ºã\203©ã\203\225ã\201®é\232£æ\216¥ã\203ªã\202¹ã\203\210
        rG[to].push_back(from);//ä¸\216ã\201\210ã\202\211ã\202\214ã\201\237ã\202ºã\203©ã\203\225ã\201®ç\237¢å\215°ã\202\222é\200\206ã\201\227ã\201\237æ\234\211å\220\221ã\202ºã\203©ã\203\225ã\201®é\232£æ\216¥ã\203ªã\202¹ã\203\210
}
//ä¸\200å°¦ç\233®ã\201®dfs
void dfs(int v){
        used[v] = true;
        for (int i = 0; i < G[v].size(); ++i){
                if(!used[G[v][i]]) dfs(G[v][i]);
        }
        vs.push_back(v);//ã\201\223ã\202\214ä»¥ä¸\212é\200²ã\202\201ã\201ªã\201\217ã\201ªã\201£ã\201\237ã\202\202ã\201®ã\201\213ã\202\211é \206ã\201«vsã\201«é \202ç\202¹ç\225ªå\217•ã\202\222å\205¥ã\202\214ã\201¦ã\201\204ã\201\217
}
//2å°¦ç\233®ã\201®dfs
void rdfs(int v, int k){
        used[v] = true;
        cmp[v] = k;//é \202ç\202¹vã\201«å¯¾ã\201\227ã\201¦ã\200\201kç\225ªç\233®ã\201¨å¼•é\200£çµ\220æ\210\220å\210\206ã\201§ã\201\202ã\202\213ã\201\223ã\201¨å\205¥ã\202\214ã\202\213
        for (int i = 0; i < rG[v].size(); ++i){
                if(!used[rG[v][i]]) rdfs(rG[v][i], k);
        }
}
int scc(){
        memset(used, 0, sizeof(used));//0(ä½¿ã\201£ã\201¦ã\201ªã\201\204)ã\201§å\210\235æ\234\237å\214\226
        vs.clear();//å\210\235æ\234\237å\214\226
        for (int v = 0; v < V; ++v){
                if(!used[v]) dfs(v);
        }
        memset(used, 0 , sizeof(used));
        int k = 0;//å¼•é\200£çµ\220æ\210\220å\210\206ã\202\222å\210\206ã\201\221ã\202\213ç\225ªå\217•
        for (int i = vs.size() - 1; i >= 0; --i){//vsã\201«å\205¥ã\201£ã\201¦ã\201\204ã\202\213å¾\214ã\202\215ã\201®ã\202\202ã\201®ã\201\213ã\202\211dfs
                if(!used[vs[i]]){
                        rdfs(vs[i], k); k++;
                }
        }
        return k;//å¼•é\200£çµ\220æ\210\220å\210\206ã\201®æ\225°
}


////////////////////////////////////////////////
////////////// ./graph/scc_and_twosat.h ////////////
////////////////////////////////////////////////

class strongly_connected_components{
public:
```

```
1443:         int group_cnt; // sccã\201®æ\225°
1444:         vector<vector<int> > G, rG;
1445:         vector<int> used, vs;
1446:         vector<int> cmp; //cmp[v] := é \202ç\202¹vã\201\214å\220«ã\201¾ã\202\214ã
\202\213é\200£çµ\220æ\210\220å\210\206ã\201\214ã\201©ã\202\214ã\201ªã\201®ã\201\213ã\202\222
ç¤ºã\201\231ç\225ªå\217•
1447:         strongly_connected_components(const vector<vector<int> > &g, const vector<ve
ctor<int> > &rg, int n):
1448:                 G(g), rG(rg), cmp(2 * n), used(2 * n){
1449:                 //mainã\201®å\207¦ç\220\206
1450:                 fill(used.begin(), used.end(), 0);
1451:                 for (int i = 0; i < G.size(); ++i){
1452:                         if(!used[i]) dfs(i);
1453:                 }
1454:                 fill(used.begin(), used.end(), 0);
1455:                 int k = 0;
1456:                 for (int i = vs.size() - 1; i >= 0; --i){
1457:                         if(!used[vs[i]]) rdfs(vs[i], k++);
1458:                 }
1459:                 group_cnt = k;
1460:         }
1461:         int operator[](int i){//é\200£çµ\220æ\210\220å\210\206ã\201®ç\225ªå\217•ã
\202\222è¿\224ã\201\231
1462:                 return cmp[i];
1463:         }
1464: private:
1465:         void dfs(int curr){
1466:                 used[curr] = true;
1467:                 for(auto next : G[curr]){
1468:                         if(!used[next]) dfs(next);
1469:                 }
1470:                 vs.push_back(curr);
1471:         }
1472:         void rdfs(int curr, int k){
1473:                 used[curr] = true;
1474:                 cmp[curr] = k;//é \202ç\202¹vã\201«å¯¾ã\201\227ã\201¦ã\200\201kç\225
ªç\233®ã\201¨å¼•é\200£çµ\220æ\210\220å\210\206ã\201§ã\201\202ã\202\213ã\201\223ã\201¨å\205¥ã
\202\214ã\202\213
1475:                 for(auto next : rG[curr]){
1476:                         if(!used[next]) rdfs(next, k);
1477:                 }
1478:         }
1479: };
1480:
1481: class twosatisfiability{
1482: public:
1483:         int V;
1484:         vector<int> res; // 1:= 0:=
1485:         vector<vector<int> > g, rg;
1486:         twosatisfiability(int n) : V(n), g(2 * n), rg(2 * n), res(n){}
1487:
1488:         bool exec() {
1489:                 strongly_connected_components scc(g, rg, V);
1490:                 for (int i = 0; i < V; i++) {
1491:                         if (scc[i] == scc[i + V]) return false;
1492:                         res[i] = scc[i] > scc[i + V];
1493:                 }
1494:                 return true;
1495:         }
1496:         void add_edge(int a, int b){
1497:                 g[a].push_back(b);
1498:                 rg[b].push_back(a);
1499:         }
1500:         //0~V-1: x_i
1501:         //V~2V-1: notx_i
1502:         void add(int a, bool apos, int b, bool bpos){//a V b ã\202\222ã\202°ã\203©ã
\203\225ã\201¸
1503:                 add_edge(a + (apos ? V : 0), b + (bpos ? 0 : V)); // not a -> b
1504:                 add_edge(b + (bpos ? V : 0), a + (apos ? 0 : V)); // not b -> a
1505:         }
1506:         bool operator[](int k){
1507:                 return res[k];
1508:         }
1509: };
1510:
1511: int main(void){
1512:
1513:         twosatisfiability sat(n); /* é \202ç\202¹ã\201®æ\225° */
1514:         /* ã\201\235ã\201®ã\201¾ã\201¾ã\202\222true, å¤\211æ\233´ã\201\231ã\202\213ã
\201®ã\202\222falseã\201¨ã\201\227ã\201¦
1515:                 æ\235¡ä»¶ã\202\222æº\200ã\201\237ã\201\225ã\201ªã\201\204ã\202\210ã
\201\206ã\201ªã\202\202ã\201®ã\201®æ\235¡ä»¶ã\202\222å\217\215è»¢ã\201\225ã\201\233ã\201\237
è¾¼ºã\202\222è¿½å\212 ã\201\231ã\202\213
1516:                 ex) true trueã\201§æ\235¡ä»¶ã\202\222æº\200ã\201\237ã\201\225ã\201ªã
\201\204ã\201ªã\202\211
1517:                 sat.add(i, false, j, false)
1518:         */
1519:         return 0;
1520: }
1521:
1522: /*
1523: http://yukicoder.me/problems/no/470
1524: http://yukicoder.me/problems/no/274
1525: http://yukicoder.me/problems/no/483
1526: http://codeforces.com/contest/776/problem/D
1527: */
1528:
1529: ##################################################
1530: ############## ./graph/scc_class.h ###############
1531: ##################################################
1532:
1533: class strongly_connected_components{
1534: public:
1535:         vector<vector<int> > G, rG;
1536:         vector<int> used, vs;
1537:         vector<int> cmp; //cmp[v] := é \202ç\202¹vã\201\214å\220«ã\201¾ã\202\214ã
\202\213é\200£çµ\220æ\210\220å\210\206ã\201\214ã\201©ã\202\214ã\201ªã\201®ã\201\213ã\202\222
ç¤ºã\201\231ç\225ªå\217•
1538:         strongly_connected_components(int n): G(n), rG(n), cmp(n), used(n){}
1539:         void add_edge(int from, int to){
1540:                 G[from].push_back(to);
1541:                 rG[to].push_back(from);
1542:         }
1543:         int scc(){
1544:                 fill(used.begin(), used.end(), 0);
1545:                 vs.clear();
1546:                 for (int i = 0; i < G.size(); ++i){
1547:                         if(!used[i]) dfs(i);
1548:                 }
1549:                 fill(used.begin(), used.end(), 0);
1550:
1551:                 int k = 0;
1552:                 for (int i = vs.size() - 1; i >= 0; --i){
1553:                         if(!used[vs[i]]){
1554:                                 rdfs(vs[i], k++);
1555:                         }
1556:                 }
1557:                 return k; //sccã\201®æ\225°
1558:         }
1559: private:
1560:         void dfs(int curr){
1561:                 used[curr] = true;
```

```
1562:                 for(auto next : G[curr]){
1563:                         if(!used[next]) dfs(next);
1564:                 }
1565:                 vs.push_back(curr);
1566:         }
1567:         void rdfs(int curr, int k){
1568:                 used[curr] = true;
1569:                 cmp[curr] = k;//é \202ç\202¹vã\201«å¯¾ã\201\227ã\201¦ã\200\201kç\225ªç\233®ã\201¨å¼•é\200£çµ\220æ\210\220å\210\206ã\201§ã\201\202ã\202\213ã\201\223ã\201¨å\205¥ã\202\214ã\202\213
1570:                 for(auto next : rG[curr]){
1571:                         if(!used[next]) rdfs(next, k);
1572:                 }
1573:         }
1574: };
1575:
1576: ////////////////////////////////////////////////
1577: ////////// ./math/combination_samll.cpp //////////
1578: ////////////////////////////////////////////////
1579:
1580: const int MAX_N = 60;
1581: ll com[MAX_N][MAX_N];
1582: void combination(void){ //com[i][j] := iCj
1583:         com[0][0] = 1;
1584:         for (int i = 1; i <= MAX_N - 1; ++i){
1585:                 for (int j = 0; j <= i; ++j){
1586:                         //ã\203\221ã\202¹ã\202«ã\203«ã\201®ï¼\223è§\222å½¢
1587:                         if(j == 0) com[i][j] = com[i - 1][j];
1588:                         else com[i][j] = (com[i - 1][j] + com[i - 1][j - 1]);
1589:                 }
1590:         }
1591: }
1592:
1593: /*
1594: http://abc057.contest.atcoder.jp/tasks/abc057_d
1595: (http://abc057.contest.atcoder.jp/submissions/1184866)
1596: */
1597:
1598: ////////////////////////////////////////////////
1599: ////////////// ./math/combinations.h //////////////
1600: ////////////////////////////////////////////////
1601:
1602: const int MAX_N = 2000000;
1603: ll inv[MAX_N + 10];
1604: ll fac[MAX_N + 10], facInv[MAX_N + 10];
1605: class MATH{
1606: public:
1607:         MATH(){
1608:                 inverse();
1609:                 factroial();
1610:         }
1611:         ll nCk(ll n, ll k){// n! / k!*(n-k)!
1612:                 if(k < 0 || k > n) return 0;
1613:                 ll ret = fac[n];
1614:                 (ret *= facInv[k]) %= MOD;
1615:                 (ret *= facInv[n - k]) %= MOD;
1616:                 return ret;
1617:         }
1618:         ll nHk(ll n, ll k){// nHk = n+k-1 C k = (n+k-1)! / k! * (n-1)!
1619:                 if(n == 0 && k == 0) return 1;
1620:                 ll ret = fac[n + k - 1];
1621:                 (ret *= facInv[k]) %= MOD;
1622:                 (ret *= facInv[n - 1]) %= MOD;
1623:                 return ret;
1624:         }
1625:         ll nPk(ll n, ll k){//nPk = n! / (n-k)!
1626:                 if(k < 0 || k > n) return 0;
1627:                 ll ret = fac[n];
1628:                 (ret *= facInv[n - k]) %= MOD;
1629:                 return ret;
1630:         }
1631: private:
1632:         void inverse(void){
1633:                 inv[1] = 1;
1634:                 for (int i = 2; i <= MAX_N; ++i){
1635:                         // inv[i] = MOD - (MOD / i) * inv[MOD % i] % MOD;
1636:                         inv[i] = inv[MOD % i] * (MOD - MOD / i) % MOD;
1637:                 }
1638:         }
1639:         void factroial(void){
1640:                 fac[0] = facInv[0] = 1;
1641:                 for (int i = 1; i <= MAX_N; ++i){
1642:                         fac[i] = (fac[i - 1] * i) % MOD;
1643:                         facInv[i] = (facInv[i - 1] * inv[i]) % MOD;
1644:                 }
1645:         }
1646: };
1647:
1648: ////////////////////////////////////////////////
1649: /////////////// ./math/eratos.h /////////////////
1650: ////////////////////////////////////////////////
1651:
1652: bool isprime[20010];
1653: //ã\202¨ã\203©ã\203\210ã\202¹ã\203\206ã\203\215ã\202¹
1654: void eratos(int m){
1655:     for (int i = 0; i <= m; ++i) isprime[i] = true;
1656:     isprime[0] = isprime[1] = false;
1657:     //iã\202\222æ®\213ã\201\227ã\201¦iã\201®å\200\215æ\225°ã\202\222æ¶\210ã\201\227ã\201¦ã\201\204ã\201\217
1658:     for (int i = 2; i <= m; ++i){
1659:         if(isprime[i]){
1660:             for (int j = 2 * i; j <= m; j += i){
1661:                 isprime[j] = false;
1662:             }
1663:         }
1664:     }
1665: }
1666:
1667: ////////////////////////////////////////////////
1668: /////////////// ./math/invmod.hpp /////////////////
1669: ////////////////////////////////////////////////
1670:
1671: // x^k (mod m)
1672: long long powmod(long long x, long long k, long long m){
1673:         long long ret = 1;
1674:         while(k){
1675:                 if(k & 1) ret = ret * x % m;
1676:                 x = x * x % m;
1677:                 k >>= 1;
1678:         }
1679:         return ret;
1680: }
1681: // 1/a mod(pçˊæ\225°))
1682: long long invmod(long long a, long long p){
1683:         return powmod(a, p - 2, p);
1684: }
1685:
1686: /*
1687: ã\203\225ã\202§ã\203«ã\203\236ã\203¼ã\201®å®\232ç\220\206ã\202\222ç\224¨ã\201\204ã\201¦, modã\201®é\200\206å\205\203
1688: http://tubo28.me/algorithm/modinv/
1689:
```

```
1690:
1691: http://yukicoder.me/problems/no/492
1692: (http://yukicoder.me/submissions/157384)
1693: */
1694:
1695: ################################################
1696: ############### ./math/matrix.cpp ###############
1697: ################################################
1698:
1699: typedef vector<long long> vec;
1700: typedef vector<vec> mat;
1701: mat mul(mat &A,mat &B){ // A * B ã\201®è¨\210ç®\227
1702:         mat C(A.size(), vec(B[0].size()));
1703:         for(int i = 0; i < A.size(); i++){
1704:                 for(int k = 0;k < B.size(); k++){
1705:                         for(int j = 0; j < B[0].size(); j++){
1706:                                 C[i][j] = (C[i][j] + A[i][k] * B[k][j]) % (ll)MOD; /
/ mod
1707:                         }
1708:                 }
1709:         }
1710:         return C;
1711: }
1712: mat pow(mat A, long long n){ // A^n ã\201®è¨\210ç®\227 O(m^3*logn)
1713:         mat B(A.size(), vec(A.size()));
1714:         for(int i = 0; i < A.size(); i++){
1715:                 B[i][i] = 1;
1716:         }
1717:         while(n > 0){
1718:                 if(n & 1) B = mul(B, A);
1719:                 A = mul(A, A);
1720:                 n >>= 1;
1721:         }
1722:         return B;
1723: }
1724:
1725: int main(void){
1726:         ll n; cin >> n;
1727:         mat A(2, vec(2));
1728:         A[0][0] = 100, A[0][1]  = 1;
1729:         A[1][0] = 0,    A[1][1] = 1;
1730:         auto ret = pow(A, n - 1);
1731: }
1732:
1733:
1734: /*
1735: è¡\214å\210\227æ¼\224ç®\227 O(m^3*logn)
1736: A^n è¡\214å\210\227ã\202\222è§£ã\201\217
1737: http://yukicoder.me/problems/no/492
1738: (http://yukicoder.me/submissions/157402)
1739:
1740: http://abc009.contest.atcoder.jp/tasks/abc009_4
1741:
1742:  N*Nè¡\214å\210\227Aã\202\222ä½\234ã\202\212ã\201\237ã\201\204å ´å\220\210
1743:  mat A(N, vec(N));
1744:
1745: */
1746:
1747: ################################################
1748: ############### ./math/powmod.h ################
1749: ################################################
1750:
1751: //x^k mod
1752: long long powmod(long long x, long long k, long long m){ // ç¹°ã\202\212è¿\224ã\201
\227ä°\214ã¹\227æ³\225(logk)
1753:         if(k == 0) return 1;
1754:         if(k % 2 == 0) return powmod(x * x % m, k / 2, m);
1755:         else return x * powmod(x, k - 1, m) % m;
1756: }
1757:
1758: /*
1759:  http://arc044.contest.atcoder.jp/tasks/arc044_b
1760:  (http://arc044.contest.atcoder.jp/submissions/1151541)
1761: */
1762:
1763:
1764: ################################################
1765: ########## ./math/prime_factorization.h ##########
1766: ################################################
1767:
1768: class primelib{
1769: public:
1770:         vector<int> prime; //prime[i] := iã\201\214ç´ æ\225°ã\201®å ´å\220\210ã\201«
ã\201¯0, ä»¥å¤\226ã\201®ã\201¨ã\201\215ã\201¯iã\202\222å\211²ã\202\212ã\202\213æ\234\200å¤§ã
\201®ç´ æ\225°
1771:         vector<int> memo;
1772:         primelib(int size):prime(size + 10, 0), memo(size + 10, 0){
1773:                 prime_factorization(size + 1);
1774:         }
1775:         //[left,right]ã\201¾ã\201§ã\201®lcmã\201\214memo[x]:=xã\201®æ¬¡æ\225° ã\201¨
ã\201\227ã\201¦è¡¨ã\201\225ã\202\214ã\202\213
1776:         void getlcm(int left, int right){
1777:                 // fill(memo.begin(), memo.end(), 0); //0ã\201§å\210\235æ\234\237å
\214\226
1778:                 for (int i = max(2, left); i <= right; ++i){
1779:                         map<int, int> degree; //<x, k> := x^k
1780:                         int tmp = i;
1781:                         while(prime[tmp]){
1782:                                 degree[prime[tmp]]++;
1783:                                 tmp /= prime[tmp]; //æ\234\200å¤§ã\201®ç´ æ\225°ã
\201§å\211²ã\202\213
1784:                         }
1785:                         degree[tmp]++; //æ®\213ã\201£ã\201\237ç´ æ\225°ã\202\222ã
\202«ã\202¦ã\203³ã\203\210
1786:                         //memo[x] := [left, right]ã\201¾ã\201§ã\202\222ç´ å\233 æ
\225°å\210\206è§£ã\201\227ã\201\237æ\234\200å¤§ã\201®xã\201®æ¬¡æ\225°ã\202\222æ\233´æ\226°
1787:                         for(auto u : degree){
1788:                                 memo[u.first] = max(memo[u.first], u.second);
1789:                         }
1790:                 }
1791:         }
1792: private:
1793:         //prime[i] := iã\201\214ç´ æ\225°ã\201®å ´å\220\210ã\201«ã\201¯0, ä»¥å¤\226ã
\201®ã\201¨ã\201\215ã\201¯iã\202\222å\211²ã\202\212ã\202\213æ\234\200å¤§ã\201®ç´ æ\225°
1794:         //ã\201\223ã\202\214ã\202\222ä½¿ã\201\210ã\201°O(logn)ã\201§ç´ å\233 æ\225°å
\210\206è§£å\217¯è\203½
1795:         void prime_factorization(int n){
1796:         for (int i = 2; i <= n; i++) {
1797:                 if (prime[i] == 0) {
1798:                         for (int j = 2; j * i <= n; j++) {
1799:                         prime[i * j] = i;
1800:                         }
1801:                 }
1802:         }
1803:     }
1804: };
1805:
1806: ################################################
1807: ############### ./other/is_ururu.h ###############
1808: ################################################
1809:
1810: bool is_ururu(int y){
```

```
1811:         if(y % 4) return false;
1812:         else if(y % 100) return true;
1813:         else if(y % 400) return false;
1814:         else return true;
1815: }
1816:
1817: ################################################
1818: ############# ./search/rollinghash.h #############
1819: ################################################
1820:
1821: typedef unsigned long long ull;
1822: const ull B = 100000007;
1823: //aã\201¯ï½\202ã\201«ã\201\204ã\201\217ã\201¤
1824: int contain(string a, string b){
1825:     int ret = 0;
1826:     int al = a.length(), bl = b.length();
1827:     if(al > bl) return 0;
1828:
1829:     ull t = 1;
1830:     for (int i = 0; i < al; ++i) t *= B;
1831:
1832:     ull ah = 0, bh = 0;
1833:     for (int i = 0; i < al; ++i) ah = ah * B + a[i];
1834:     for (int i = 0; i < al; ++i) bh = bh * B + b[i];
1835:
1836:     for (int i = 0; i + al <= bl; ++i){
1837:         if(ah == bh) ret++;
1838:         if(i + al < bl) bh = bh * B + b[i + al] - b[i] * t;
1839:     }
1840:     return ret;
1841: }
1842:
1843: /*
1844: http://yukicoder.me/problems/no/430
1845: */
1846:
1847:
1848:
1849: ################################################
1850: ############### ./search/xor128.h ################
1851: ################################################
1852:
1853: unsigned int xor128(){//ä¹±æ\225°ç\224\237æ\210\220
1854:         static unsigned int x = 123456789, y = 362436069, z = 521288629, w = 8867512
3;
1855:         unsigned int t = (x ^(x << 11));
1856:         x = y; y = z; z = w;
1857:         return (w = (w ^ (w >> 19)) ^ (t ^ (t >> 8)));
1858: }
1859:
1860: /*
1861: http://yukicoder.me/problems/no/469
1862: */
1863:
```