```
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:
        32: ########## ./data struture/imos one.h ###########
        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){</pre>
        40:
                                       imos[i] = imos[i - 1] + baseimos[i];
        41:
        42:
                            return imos;
        43: }
        44:
        46: #### ./data struture/segtree lazy add min.cpp ####
        åº\217å\200¤
        50: struct segtree{
        51: public:
        52:
                             const int SIZE = 1 << 18;</pre>
                             //seg: å\214° \(\ell \) 226\223 \(\alpha\) 201\(\ell \) \(\alpha\) 234\200\(\alpha\) \(\alpha\) 217\(\alpha\) \(\alpha\) 200\(\alpha\) \(\alpha\) \(\alpha\) 214° \(\ell \) \(\alpha\) 226\223 \(\alpha\) \(\alp
ã\201/ã\201\204ã\202\213ã\202\202ã\201®
                             \label{eq:continuous} $$ \214^{\circ} (226 \223 \201^{\circ})^{\circ} (220 \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
        56:
\224"æ\2261æ3\225
        57:
                                       if(lazy[k] != 0){
                                                  seg[k] += lazy[k];//a^214°e^226^223[1,r)a^201~a^201^231a^201^1a^201/a^220
```

```
59:
                                                                                                                               if(r - 1 > 1)
                     60:
                                                                                                                                                            lazy[k * 2 + 1] += lazy[k]; //é \langle 201 \rangle 205å * \sqrt{3} \langle 202 \rangle 222å \cdot /3 \langle 201 \otimes a - \rangle 2203
   \201«ä¼\235æ\220¬
                     61:
                                                                                                                                                            lazy[k * 2 + 2] += lazy[k]; //é(201(205å)*¶ã(202(222å(217³ã(201@å-(220))*)); //é(201(205å)*¶ã(202(222å(217³ã(201@å-(220)))); //é(201(205å)*¶ã(202(222å(217³ã(201@å-(220)))); //é(201(205å)*¶ã(202(222å(217³ã(201@å-(220)))); //é(201(205å)*¶ã(202(222å(217³ã(201@å-(220)))); //é(201(205å)*¶ã(202(222å(217³ã(201@å-(220)))); //é(201(205å)*¶ã(202(222å(217³ã(201@å-(220)))); //é(201(205å)); //é(201(205å)); //é(201(205å)); //é(201(205å)); //é(201(205å)); //é(201(205å)); //é(201(205å)); //é(201(205å)); //é(201(205á)); //é(200(205á)); //é(200(205á)); //é(200(205á)); //é(200(205á)); //é(200(205á)); //é(200(2056)); //é(200(2056)); ///é(200(2056)); ///é(200(2056)); ///é(200(2056)); ///é(200(2056)); ///é(200(2056)); ///é(200(2056)); ///é(200(2056)); ///é(200(2056)); ///é(200(2056)); ///é(2
 ã\201«ä¼\235æ\220¬
                     62:
                     63:
                                                                                                                                lazy[k] = 0; //\tilde{a} 203 \ 216\tilde{a} \ 203 \ 211k\tilde{a} \ 201 \ \ddot{a} \ 223 \ as \ 220 \ \ddot{a} \ 2214 \ \ddot{a} \ 206
                     64:
                     65:
                     66:
                                                                           void update(int a, int b, int k, int l, int r, int x){
                     67:
                                                                                                      lazv evaluate(k, l, r);
                                                                                                       if(r <= a || b <= 1) return;</pre>
                     68:
                     69:
                                                                                                      if(a <= 1 && r <= b){
                     70:
                                                                                                                                lazy[k] += x; //a^{212} \tilde{a} 201 210\tilde{a} 202 213
                     71:
                                                                                                                                lazv evaluate(k, l, r);
                     72:
                                                                                                      }else{
                                                                                                                                update(a, b, k * 2 + 1, 1, (1 + r) / 2, x);
                     73:
                     74:
                                                                                                                                update(a, b, k * 2 + 2, (1 + r) / 2, r, x);
                     75:
                                                                                                                                seg[k] = min(seg[k * 2 + 1], seg[k * 2 + 2]); //a\214°e\226\223ã\201@min
                     76:
                     77:
                     78:
                                                                          11 query(int a, int b, int k, int l, int r){
                     79:
                                                                                                      lazy evaluate(k, l, r);
                     80:
                                                                                                      if(r <= a | | b <= 1) return INF;//minã\201@å½±é\237;ã\201@ã\201ªã\201\204ã
   \202\202ã\201®
                     81:
                                                                                                     if(a <= 1 && r <= b) return seq[k];</pre>
                     82:
                                                                                                      11 \times = query(a, b, k * 2 + 1, 1, (1 + r) / 2);
                     83:
                                                                                                      11 y = query(a, b, k * 2 + 2, (1 + r) / 2, r);
                     84:
                                                                                                      return min(x, y); //å•/å\217³ã\201@minã\202\222
                     85:
                     86:
                                                                          //update(a,b,x) := [a,b)\tilde{a}\langle 202\langle 222a^2\langle 205\tilde{a}\langle 201|x\tilde{a}\langle 202\langle 222a^2\langle 212|\tilde{a}\langle 201\langle 210\tilde{a}\langle 202\langle 213|\tilde{a}\langle 201|x\tilde{a}\langle 202|\tilde{a}\langle 212|\tilde{a}\langle 212|\tilde{a}
                     87:
                                                                          void update(int a, int b, int x){update(a, b, 0, 0, SIZE, x);}
                     88:
                                                                          //query(a,b) := [a,b)\tilde{a} 201 < a^{-1}\tilde{a} 201 < 231\tilde{a} 202 < 213\tilde{a} 203 < 201\tilde{a} 200\tilde{a} 201\tilde{a} 200\tilde{a} 202 < 222\tilde{a} \pm 231\tilde{a} 201\tilde{a} 201\tilde{a}
   \202ã\202\201ã\202\213
                     89:
                                                                          11 guery(int a, int b){return guery(a, b, 0, 0, SIZE);}
                     90: };
                     91:
                     92: /*
                     åº\217å\200¤
                     94: const int MAX N = 1 << 19;
                     95: struct seatree(
                     96: public:
                     97:
                                                                                                      int seg[MAX_N * 2], lazy[MAX_N * 2];//segã\201~欲ã\201\227ã\201\204æ\203
   \205a ± lazyã\201-å\214°é\226\223ã\201«å-¾ã\201\231ã\202\213ä,\200æ§\230ã\201ªå\207/c\220
   \206ã\202\222礰ã\201\231ã\202\202ã\201®
                                                                         //seg:å\214°é\226\223ã\201®æ\234\200å°\217å\200¤ lazy:å\214°é\226\223ã\201«å<sup>-</sup>¾ã
   \201\227ã\201\ã\200\201å\212 ã\201\210ã\202\213å\200¤ã\201§ã\201¼ã\201 é\201\205å»¶ã\201\227
  ã\201/ã\201\204ã\202\213ã\202\202ã\201®
             100:
                                                                         segtree(int n){
               101:
                                                                                                      SIZE = 1;
               102:
                                                                                                      \227ã\201«
                                                                                                       for (int i = 0; i < 2 * SIZE - 1; ++i) seg[i] = lazy[i] = 0;//0<math>\tilde{a}\201\tilde{e}\201
   \205a*¶ã\201\227ã\201|ã\201\204ã\202\213ã\202\202ã\201®ã\201\214ã\201*ã\201\204ã\201\223ã
  \201"ã\202\222礰ã\201\231
          104:
             105:
                                                                          void lazy_evaluate(int k, int 1, int r){//6 \ 201 \ 205å \ \#203 \ 205å \ \pm \~a \ 201@6 \ 201@c}
   \224"æ\2261æ3\225
            106:
                                                                                                      if(lazy[k] != 0){
              107:
                                                                                                                                 \214ã\201\230å\200¤ã\202\222追å\212 ã\201\231ã\202\213ã\201\223ã\201"ã\201«ã\201°ã\201£ã
   201/\tilde{a}201/204\tilde{a}201/\tilde{a}200/201seg\tilde{a}201\tilde{a}201-\tilde{a}201-\tilde{a}200d^{2}201d^{2}201/21d^{2}201/21d^{2}201f^{2}201/21d^{2}201f^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}201/21d^{2}2
\tilde{a} \setminus 201 \setminus 204\tilde{a} \setminus 202 \setminus 213\tilde{a} \setminus 201 \cdot \tilde{a} \setminus 201 \cdot \tilde{a} \setminus 201 \cdot \tilde{a} \setminus 201 \setminus 210\tilde{a} \setminus 202 \setminus 213\tilde{a} \setminus 201 \cdot \tilde{a} \setminus 201 \setminus 210\tilde{a} \setminus 202 \setminus 213\tilde{a} \setminus 201 \setminus 210\tilde{a} \setminus 210\tilde{a}
```

ã\201\204ã\202\213ã\201®ã\201\8ã\200\201å\212 ã\201\210ã\202\213å\200¤ã\202\222è¶³ã\201\231

```
108:
                                                                                                                                                                              if(r - 1 > 1){
                 109:
                                                                                                                                                                                                                  lazy[k * 2 + 1] += lazy[k];//é(201(205å)*¶ã(202(222å)*[ã(201@å-(220ã)*]*]
 \201«ä¼\235æ\220¬
                110:
                                                                                                                                                                                                                   lazy[k * 2 + 2] += lazy[k]; //é (201 (205å) ¶ã(202 (222å) (217³ã(201) @å-(220) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (205) (20
ã\201«ä¼\235æ\220¬
                 111:
                 112:
                                                                                                                                                                                lazy[k] = 0; //\tilde{a} \ 203 \ 216\tilde{a} \ 203 \ 211k\tilde{a} \ 201 \ \tilde{a} \ 2207 \ \tilde{a} \ 2207 \ \tilde{a} \ 214\tilde{a} \ 206 \ 214\tilde{a} \ 206 \ 214\tilde{a} \ 206 \ 214\tilde{a} \ 207 \ \tilde{a} \ 207 \
                 113:
                 114:
                   115:
                                                                                                       void update(int a, int b, int k, int l, int r, int x){
                   116:
                                                                                                                                         lazy evaluate(k, 1, r);
                 117:
                                                                                                                                          if(r <= a | | b <= 1) return;
                   118:
                                                                                                                                         if(a <= 1 && r <= b){
                   119:
                                                                                                                                                                              lazy[k] += x; //å 212 ~\tilde{a} 201 210~\tilde{a} 202 213
                   120:
                                                                                                                                                                                lazy evaluate(k, l, r);
                   121:
                                                                                                                                          }else{
                   122:
                                                                                                                                                                              update(a, b, k * 2 + 1, 1, (1 + r) / 2, x);
                   123:
                                                                                                                                                                              update(a, b, k * 2 + 2, (1 + r) / 2, r, x);
                   124:
                                                                                                                                                                                seg[k] = min(seg[k * 2 + 1], seg[k * 2 + 2]); //å\214°é\226\223ã\201@min
                   125:
                   126:
                   127:
                                                                                                       11 query(int a, int b, int k, int 1, int r){
                   128:
                                                                                                                                          lazy evaluate(k, 1, r);
                 129:
                                                                                                                                          \202\202ã\201®
                                                                                                                                         if(a <= 1 && r <= b) return seg[k];</pre>
                 130:
                 131:
                                                                                                                                          11 \times = query(a, b, k * 2 + 1, 1, (1 + r) / 2);
                   132:
                                                                                                                                          11 \ y = query(a, b, k * 2 + 2, (1 + r) / 2, r);
                   133:
                                                                                                                                          return min(x, y); //å \cdot |å \cdot 217³ã \cdot 201@minã \cdot 202 \cdot 222
                 134:
                   135:
                                                                                                       //update(a,b,x) := [a,b)\tilde{a}\202\222a\205\tilde{a}\201/x\tilde{a}\202\222a\202\222a\201/x\tilde{a}\201\210\tilde{a}\202\213
                   136:
                                                                                                     void update(int a, int b, int x){update(a, b, 0, 0, SIZE, x);}
                 137:
                                                                                                       //query(a,b) := [a,b)\tilde{a} 201 \leq \tilde{a}^{*}\tilde{a} 201 \leq \tilde{a}^{*}\tilde{a} 201 \leq \tilde{a}^{*}\tilde{a} 201 \leq \tilde{a}^{*}\tilde{a} \leq \tilde{a}^
    \202ã\202\201ã\202\213
                   138:
                                                                                                       11 query(int a, int b){return query(a, b, 0, 0, SIZE);}
                 139: };
                 140: */
                 141:
                 143: http://tenka1-2016-qualb.contest.atcoder.jp/tasks/tenka1_2016_qualB_d
                 144: */
                 145:
                   148: ##### ./data struture/segtree lazv add min.h #####
                 151: //(1)å\214°é\226\223ã\201«ä,\200æ§\230å\212 ç®\227 (2)å\214°é\226\223ã\201®æ\234\200
 å°\217å\200¤
                 152: struct segtree{
                 153: public:
                                                                                                     const int SIZE = 1 << 18;
                                                                                                        \begin{tabular}{ll} $$ \201\227\ \align{tabular} \begin{tabular}{ll} $$ \201\227\ \align{tabular} \begin{tabular}{ll} $$ \201\227\ \align{tabular} \begin{tabular}{ll} \end{tabular} \begin{tabular}{ll} \end{t
\tilde{a} \ 201/\tilde{a} \ 201 \ 204\tilde{a} \ 202 \ 213\tilde{a} \ 202 \ 202\tilde{a} \ 2010
                                                                                                     (214^{\circ})^{2}(226^{2})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(201^{\circ})^{2}(20
\231ã\202\202ã\201®
                157:
                                                                                                       segtree():seg(SIZE * 2), lazy(SIZE * 2){}
                 158:
                                                                                                     void lazy evaluate(int k, int l, int r)\{//6 \setminus 201 \setminus 205\text{å} \times \text{ma} \setminus 203 \setminus 205\text{å} \pm \text{a} \setminus 201 \text{me} \in 201 \text{me}
 \224"æ\2261æ3\225
                159:
                                                                                                                                         if(lazy[k] != 0){
                                                                                                                                                                                seg[k] += lazy[k]; //å \langle 214°\acute{e} \langle 226 \langle 223[1,r) \tilde{a} \langle 201 \langle \tilde{a} \langle 201 \rangle \langle 231\tilde{a} \langle 201 \rangle \langle 231\tilde{a} \rangle \langle 201 \rangle \langle 231\tilde{a} \langle 201 \rangle \langle 231\tilde{a} \rangle \langle 231\tilde{a} \rangle \langle 231\tilde{a} \langle 231 \rangle \langle 231\tilde{a} \rangle \langle 231\tilde{a} \langle 231 \rangle \langle 231\tilde{a} \rangle \langle 231\tilde{a} \rangle \langle 231\tilde{a} \langle 231 \rangle \langle 231\tilde{a} \rangle \langle 231\tilde{a}
                 160:
 \214ã\201\230å\200¤ã\201\232迼å\212 ã\201\231ã\202\213ã\201\223ã\201"ã\201«ã\201ªã\201£ã
 \tilde{a} \setminus 201 \setminus 204\tilde{a} \setminus 202 \setminus 213\tilde{a} \setminus 201 \otimes \tilde{a} \setminus 201 \times \tilde{a} \setminus 201 \setminus 210\tilde{a} \setminus 202 \setminus 213\tilde{a} \setminus 201 \times \tilde{a} \setminus 201 \setminus 210\tilde{a} \setminus 202 \setminus 213\tilde{a} \setminus 201 \setminus 210\tilde{a} \setminus 201 \setminus 210\tilde{a} \setminus 201 \setminus 210\tilde{a} \setminus 201 \setminus 210\tilde{a} \setminus 210\tilde{a}
```

```
161:
                                                                                                                     if(r - 1 > 1)
              162:
                                                                                                                                               lazy[k * 2 + 1] += lazy[k]; //é < 201 < 205 å » ¶ã < 202 < 222 å • /ã < 201 @å - \ 220 ã
    \201«ä¼\235æ\220¬
           163:
                                                                                                                                              ã\201«ä¼\235æ\220¬
             164:
               165:
                                                                                                                      lazy[k] = 0; //\tilde{a} 203 \cdot 216\tilde{a} \cdot 203 \cdot \tilde{a} \cdot 201 \cdot \tilde{a} \cdot 201 \cdot \tilde{a} \cdot 2207 \cdot \tilde{a} \cdot 201 \cdot
               166:
               167:
               168:
                                                                     void update(int a, int b, int k, int l, int r, int x){
               169:
                                                                                             lazy evaluate(k, l, r);
                                                                                              if(r <= a || b <= 1) return;</pre>
               170:
               171:
                                                                                              if(a <= 1 && r <= b){
               172:
                                                                                                                      lazy[k] += x; //å 212 ~\tilde{a} 201 210~\tilde{a} 202 213
               173:
                                                                                                                      lazv evaluate(k, l, r);
               174:
                                                                                             }else{
               175:
                                                                                                                     update(a, b, k * 2 + 1, 1, (1 + r) / 2, x);
                176:
                                                                                                                     update(a, b, k * 2 + 2, (1 + r) / 2, r, x);
                177:
                                                                                                                     seg[k] = min(seg[k * 2 + 1], seg[k * 2 + 2]); //å\214°\(\delta\)23\(\alpha\)201\(\delta\)min
                178:
               179:
               180:
                                                                    11 query(int a, int b, int k, int l, int r){
               181:
                                                                                             lazy evaluate(k, l, r);
               182:
                                                                                             if(r <= a | b <= 1) return INF;//minã\201@å½±é\237;ã\201@ã\201*ã\201\204ã
    \202\202ã\201®
                                                                                             if(a <= 1 && r <= b) return seg[k];</pre>
               183:
               184:
                                                                                             11 x = query(a, b, k * 2 + 1, 1, (1 + r) / 2);
               185:
                                                                                             11 y = query(a, b, k * 2 + 2, (1 + r) / 2, r);
               186:
                                                                                             return min(x, y); //å•/å\217³ã\201@minã\202\222
                187:
               188:
                                                                     //update(a,b,x) := [a,b) \tilde{a} \ 202 \ 222 \hat{a} \ 205 \tilde{a} \ 201 \ x \tilde{a} \ 202 \ 222 \hat{a} \ 212 \tilde{a} \ 201 \ 210 \tilde{a} \ 202 \ 213
                189:
                                                                    void update(int a, int b, int x){update(a, b, 0, 0, SIZE, x);}
               190:
                                                                    //guery(a,b) := [a,b)\tilde{a} 201 < a^{4}\tilde{a} 201 < 231\tilde{a} 202 < 213x < 234 < 200a^{9} < 217a^{2} 200 = 202 < 222x = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 = 200 
    \202ã\202\201ã\202\213
               191:
                                                                    11 query(int a, int b){return query(a, b, 0, 0, SIZE);}
               192: };
               193:
               194: /*
                åº\217å\200¤
                196: const int MAX N = 1 << 19;
                197: struct segtree{
                198: public:
                199:
                                                                                              int seq[MAX N * 2], lazy[MAX N * 2];//seqã\201~x¬²ã\201\227ã\201\204æ\203
    \205a ± lazyã\201-å\214°é\226\223ã\201«å-¾ã\201\231ã\202\213ä,\200æ§\230ã\201ªå\207/ç\220
    \206ã\202\222礰ã\201\231ã\202\202ã\201®
                                                                    //seg:å\214°é\226\223ã\201®æ\234\200å°\217å\200¤ lazy:å\214°é\226\223ã\201«å¯¾ã
    \201\227\\201|\alpha\200\201\alpha\212\alpha\201\210\alpha\202\213\alpha\200\\alpha\201\\alpha\201\\alpha\201\\alpha\201\\alpha\201\\alpha\201\\alpha\201\\alpha\201\\alpha\201\\alpha\201\\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alph
   ã\201/ã\201\204ã\202\213ã\202\202ã\201®
               202:
                                                                    segtree(int n){
                203:
                204:
                                                                                             while(SIZE < n) SIZE *= 2;//è|\201c^ æ\225°ã\202\2222ã\201®ã\201¹ã\201\215ä¹
 \227ã\201«
               205:
                                                                                              for (int i = 0; i < 2 * SIZE - 1; ++i) seg[i] = lazy[i] = 0; //0ã\201^-\(\ell \)201
     \  \  \, 205\mathring{a} * \P \tilde{a} \setminus 201 \setminus 227\tilde{a} \setminus 201 / \tilde{a} \setminus 201 \setminus 204\tilde{a} \setminus 202 \setminus 213\tilde{a} \setminus 202 \setminus 202\tilde{a} \setminus 201 \\  \tilde{a} \setminus 201 \setminus 214\tilde{a} \setminus 201 \hat{a} \setminus 201 \setminus 224\tilde{a} \setminus 221 \setminus 224\tilde{a} \setminus 22
   \201"ã\202\222礰ã\201\231
               206:
                207:
                                                                    void lazy_evaluate(int k, int 1, int r){//6 \ 201 \ 205å \ \#x \ 203 \ 205å \ \pm 3 \ 201@6 \ 201@c}
    \224"æ\2261æ3\225
              208:
                                                                                             if(lazy[k] != 0){
                                                                                                                      \201\\\alpha\\201\\\204\alpha\\201\\\alpha\\alpha\\201\\\alpha\\alpha\\201\\\alpha\\\alpha\\201\\\alpha\\\alpha\\201\\\alpha\\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\\alpha\alpha\\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\\alpha\\alpha\alpha\alpha\\alpha\a
ã\201\204ã\202\213ã\201®ã\201\§ã\200\201å\212 ã\201\210ã\202\213å\200¤ã\202\222è¶³ã\201\231
               210:
                                                                                                                     if(r - 1 > 1)
```

318:

```
211:
                                                                                                                                                                                                 lazy[k * 2 + 1] += lazy[k]: //é < 201 < 205å * ¶ã < 202 < 222å • /ã < 201@å - \ 220ã
 \201«ä¼\235æ\220¬
                212:
                                                                                                                                                                                                lazy[k * 2 + 2] += lazy[k]; //é (201 (205 å) ¶ã (202 (222 å) (217 ³ã (201 @å - (220 å) 201 ) å (201 (201 å) 201 
ã\201«ä¼\235æ\220¬
                213:
                 214:
                                                                                                                                                                 lazv[k] = 0: //\tilde{a} 203 \cdot 216\tilde{a} \cdot 203 \cdot \tilde{a} \cdot 203 \cdot 211k\tilde{a} \cdot 201 - \tilde{a} \cdot (235\tilde{a} \cdot 2207) - \tilde{a} \cdot (235\tilde{a} \cdot 2207
                 215:
                 216:
                 217:
                                                                                              void update(int a, int b, int k, int l, int r, int x){
                 218:
                                                                                                                               lazv evaluate(k, l, r);
                                                                                                                               if(r <= a | | b <= 1) return;
                 219:
                 220:
                                                                                                                             if(a <= 1 && r <= b){
                   221:
                                                                                                                                                               lazv[k] += x; //å 212 ~\tilde{a} 201 210~\tilde{a} 202 213
                   222:
                                                                                                                                                                 lazy evaluate(k, 1, r);
                   223:
                                                                                                                             }else{
                   224:
                                                                                                                                                               update(a, b, k * 2 + 1, 1, (1 + r) / 2, x);
                   225:
                                                                                                                                                               update(a, b, k * 2 + 2, (1 + r) / 2, r, x);
                   226:
                                                                                                                                                               seg[k] = min(seg[k * 2 + 1], seg[k * 2 + 2]); //å\214°é\226\223ã\2018min
                   227:
                   228:
                   229:
                                                                                              11 query(int a, int b, int k, int 1, int r){
                   230:
                                                                                                                             lazy_evaluate(k, 1, r);
                 231:
                                                                                                                               if(r \le a \mid b \le 1) return INF;//minã(201@å½±é(237;ã(201@ã(201ªã(201)204ã)201)a)
  \202\202ã\201®
                 232:
                                                                                                                             if(a <= 1 && r <= b) return seg[k];</pre>
                 233:
                                                                                                                             11 \times = query(a, b, k * 2 + 1, 1, (1 + r) / 2);
                 234:
                                                                                                                             11 \ y = query(a, b, k * 2 + 2, (1 + r) / 2, r);
                 235:
                                                                                                                             return min(x, y); //å \cdot |å \cdot 217³ã \cdot 201@minã \cdot 202 \cdot 222
                 236:
                 237:
                                                                                              //update(a,b,x) := [a,b) \tilde{a} \ 202 \ 222 \ a \ 205 \ \tilde{a} \ 201 \ x \tilde{a} \ 202 \ 222 \ a \ 212 \ \tilde{a} \ 201 \ 210 \ \tilde{a} \ 202 \ 213 \ \tilde{a} \ 201 \ \tilde{a} \ 201
                 238:
                                                                                             void update(int a, int b, int x){update(a, b, 0, 0, SIZE, x);}
                 239:
                                                                                              //query(a,b) := [a,b)\tilde{a} 201 \leq \tilde{a}^{*}\tilde{a} 201 \leq \tilde{a}^{*}\tilde{a} 201 \leq \tilde{a}^{*}\tilde{a} 201 \leq \tilde{a}^{*}\tilde{a} \leq \tilde{a}^
  \202ã\202\201ã\202\213
                 240:
                                                                                             11 query(int a, int b){return query(a, b, 0, 0, SIZE);}
                 241: };
                 242: */
                 243:
                 244: /*
                 245: http://tenka1-2016-qualb.contest.atcoder.jp/tasks/tenka1 2016 qualB d
                 247:
                   250: ##### ./data struture/segtree lazy add sum.h #####
                 è"\210å\200¤
                 254: struct segtree{
                 255: public:
                 256:
                                                                                             const int SIZE = 1 << 18;</pre>
                                                                                             //seg:å\214°\\equiv \226\223\alpha\201\text{@å\220\210\alpha"\210\alpha\200\pi lazy:\alpha\214\circ\equiv \226\223\alpha\201\circ\alpha \alpha\alpha \alpha 
  ã\201/ã\201\204ã\202\213ã\202\202ã\201®
                                                                                             \label{eq:condition} $$ \214^{\circ} (226) 223 (201)^{\circ} (31)^{\circ} (201)^{\circ} (213)^{\circ} (202)^{\circ} (213)^{\circ} (201)^{\circ} (213)^{\circ} (213)^{\circ} (201)^{\circ} (213)^{\circ} (2
\231ã\202\202ã\201®
                259:
                                                                                              segtree():seg(SIZE * 2), lazy(SIZE * 2){}
                                                                                             void lazy_evaluate(int k, int l, int r){//6 \setminus 201 \setminus 205å * \# \& \setminus 203 \setminus 205å \pm \tilde{a} \setminus 201@6 \setminus 201@c}
 \224"æ\2261æ3\225
                261:
                                                                                                                             if(lazy[k] != 0){
                                                                                                                                                                 seg[k] += lazy[k]; // å \ 214° \'e \ 226 \ 223[1,r) \~a \ 201 \ \~a \ 201 \ 231\~a \ 201 \ ²\~a \ 201 \ ²\~a \ 201 \ ²\~a \ 201 \ ²\~a \ 201 \ °a \ 2
  \begin{tabular}{l} $$ \201 \aligned 201 \
263:
                                                                                                                                                              if(r - 1 > 1)
```

```
264:
                                                                     lazy[k * 2 + 1] += lazy[k];//é(201(205å)*¶ã(202(222å)*[ã(201@å-(220ã)*])** | [azy[k * 2 + 1] += lazy[k];//é(201(205å)*¶ã(202(223å)*])** | [azy[k * 2 + 1] += lazy[k];//é(201(205å)*])** | [azy[k * 2 + 1] += lazy[k];//é(201(205a))** | [azy[k * 2 + 1] += lazy[k]** | [az
\201«ä¼\235æ\220¬
     265:
                                                                     ã\201«ä¼\235æ\220¬
     266:
      267:
                                                        |\text{lazy}[k]| = 0; //\tilde{a} | 203 | 216\tilde{a} | 203 | \tilde{a} | 203 | 211k\tilde{a} | 201 | \tilde{a} | 235 | 220 | \tilde{a} | 214\tilde{a} | 206 | 214\tilde{a} | 214\tilde{
      268:
      269:
      270:
                                void update(int a, int b, int k, int l, int r, int x){
      271:
                                             lazv evaluate(k, l, r);
                                             if(r <= a || b <= 1) return;</pre>
      272:
      273:
                                             if(a <= 1 && r <= b)
      274:
                                                        lazv[k] += x; //å 212 ~a 201 210~a 202 213
      275:
                                                         lazy evaluate(k, l, r);
      276:
                                             }else{
      277:
                                                        update(a, b, k * 2 + 1, 1, (1 + r) / 2, x);
      278:
                                                        update(a, b, k * 2 + 2, (1 + r) / 2, r, x);
      279:
                                                        seg[k] = seg[k * 2 + 1] + seg[k * 2 + 2]; //å\214°\(\delta\)23\(\delta\)23\(\delta\)201\(\delta\)å\\(220\)
\210è"\210
      280:
      281:
      282:
                                11 query(int a, int b, int k, int l, int r){
      283:
                                            lazy evaluate(k, l, r);
      284:
                                            if(r <= a | b <= 1) return 0;//å\220\210e\210a\201\alpha\221\alpha\237;\alpha\201\bar@a\201\alpha
285:
                                            if(a <= 1 && r <= b) return seg[k];</pre>
      286:
                                            11 x = query(a, b, k * 2 + 1, 1, (1 + r) / 2);
      287:
                                            11 y = query(a, b, k * 2 + 2, (1 + r) / 2, r);
      288:
                                            return x + y; //å•/å\217³ã\201@å\220\210è"\210ã\202\222
      289:
      290:
                                //update(a,b,x) := [a,b) \tilde{a} \ 202 \ 222 \hat{a} \ 205 \tilde{a} \ 201 \ x \tilde{a} \ 202 \ 222 \hat{a} \ 212 \tilde{a} \ 201 \ 210 \tilde{a} \ 202 \ 213 
      291:
                                void update(int a, int b, int x){update(a, b, 0, 0, SIZE, x);}
      292:
                                //guery(a,b) := [a,b)\tilde{a} 201 < a^{4}\tilde{a} 201 < 231\tilde{a} 202 < 213\tilde{a} 200 < 210\tilde{e} 210\tilde{a} 200 = 2222\tilde{a}
\202ã\202\201ã\202\213
      293:
                                11 query(int a, int b){return query(a, b, 0, 0, SIZE);}
      294: };
      295:
      297: ####### ./data struture/segtree max.hpp ########
      299:
      300: /*
      301: å\221¼ã\201³ã\201 ã\201\227æ\226¹
      302: seqtree<int> seq(n); // intå\236\213ã\201@è|\201c´ã\201§ nã\201-è|\201c´ã\201@æ
12250
      303: */
      305: template <class T> //T : dat[]ã\201@ä,-è°«ã\201@å\236\213
      306: class segtree{
      307: public:
      308:
                                            int n;
      309:
                                            vector<T> dat;
      310:
                                             segtree(int n_): n(n_) \{ //n_e / 201c \approx 225^\circ \}
      311:
                                                                    n = 1;
     312:
                                                                     while(n < n_) n *= 2;</pre>
                                                                     \texttt{dat.resize(n * 2, 0); //(1)  å\210\235x\234\237å\200x\tilde{a}\202\222x\234}
     313:
\200a°\217ã\201« -INFã\201\213ã\202\202
     314:
     315:
                                            void update(int k, T val){ // kç\225°c\233@ã\201@å\200¤(0-indexed)ã\202\222
val ã\201«å¤\211æ\2331
    316:
                                                                     \200a\214°\eq\226\223\ai\201\ai\202\ai\203\\ai\203\\207\ai\203\\203\\202\ai\202\\ai\202\\222\ai\\213\ai\201\\213\ai\
\202\211\(\ell\) \206\(\alpha\)\210\227\(\alpha\)\231
                                                                                             dat[k>>1] = max(dat[k], dat[k ^ 1]); // (2) a\214°é\226\223ã
\201@æ\234\200大å\200¤ã\201§æ\233´æ\226°
```

Page 3 of 17

Page 4 of 17

```
319:
             320:
                                                                                                           T query(int 1, int r) { //[1, r)\tilde{a} \times 2010 \hat{a} \times 214 \hat{e} \times 226 \times 223
             321:
                                                                                                                                                                 T ret = 0; //(3) \approx 234 \times 200 \approx 3 \times 200 \approx 201 \approx 226 \approx 202 \times 201 \approx 201 \times 201 \times
å\200¤ -INFã\201\213ã\202\202
               322:
                                                                                                                                                                   for (1 += n, r += n; 1 < r; 1 >>= 1, r >>= 1)
             323:
                                                                                                                                                                                                                            if(1 \& 1) ret = max(ret, dat[1++]); //(4) å 214° é 226 223 ã
 \201@æ\234\200大å\200¤ã\201§æ\233´æ\226°
                                                                                                                                                                                                                            if(r \& 1) ret = max(ret, dat[--r]); //(4) å 214° é 226 223 ã
  325:
                326:
                                                                                                                                                                   return ret;
               327:
                328: };
                329:
               330:
               331: /*
               332: segtree ã\201\231ã\201\223ã\201\227é«\230é\200\237å\214\226
               333: æ\233´æ\226°: 1 ç\202¹
               334: ã\202~ã\202~ã\203ª: å\214°\eq\226\223ã\201@æ\234\200å°\217å\200¤
               335: note: (1)~(4)~(202\222å\211~(201\210~(202\213~(201\223~(201\223~(201\223~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201)~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201\233~(201)~(201\233~(201)~(201\233~(201)~(201)~(20
 \203%
               336: http://codeforces.com/contest/777/problem/E
               337: (http://codeforces.com/contest/777/submission/25346678)
               338: \tilde{a} \setminus 201 \setminus 223\tilde{a} \setminus 201 \otimes \hat{a} \setminus 225 \setminus 217\acute{e}_{1} \setminus 214\tilde{a} \setminus 201 \otimes \tilde{a} \setminus 201^{-1} \hat{a} \cdot \hat{a} \otimes \tilde{a} \setminus 201^{-1} \hat{a} \setminus 201 \times 201^{-1} \hat{a} \otimes \tilde{a} \setminus 201^{-1} \hat{a} \otimes \tilde{a} \otimes \tilde{a} \setminus 201^{-1} \hat{a} \otimes \tilde{a} \otimes 
               339:
               340: http://codeforces.com/contest/689/problem/D
               341: (http://codeforces.com/contest/689/submission/25354957)
               342: \tilde{a} \geq 0.1 \geq 2.3\tilde{a} \geq 0.1 = \tilde{a} \geq 0.1 = 0.1 = \tilde{a} \geq 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0.1 = 0
  \201|大ã\201\215ã\201ªå•®ã\201\214å\207°ã\201|ã\201\204ã\202\213
               343: */
               344:
               345:
                347: ###### ./data struture/segtree max idx.cpp #######
               350: template <class T> //T : dat[]ã\201@ā,-è°«ã\201@å\236\213
                351: class segtree{
               352: public:
               353:
                                                                                                            int n;
               354:
                                                                                                           T neutral;
                                                                                                            vector<T> dat;
                                                                                                         T func(T 1, T r) { //\text{å} | 214^{\circ}\text{\'e} | 226 | 223\tilde{a} | 202 | 222\tilde{a} | 203 | 236\tilde{a} | 203 | \tilde{a} | 201 | 231\tilde{a}
  \202\213\(\ellarge\)226\(\chi\)225\(\chi\)
             357:
                                                                                                                                                                   return max(1, r); /* ã\201\223ã\201\223ã\202\222å¤\211ã\201\210ã\202
 \213 */
               358:
                                                                                                            segtree(int n_, T val): n(n_), neutral(val){ //n_:e}\201c^* æ\225° val:a\215
               359:
 \230ä%\215å\205\203
             360:
                                                                                                                                                                   n = 1;
               361:
                                                                                                                                                                     while(n < n_) n *= 2;</pre>
             362:
                                                                                                                                                                   dat.resize(n * 2, neutral); //å\210\235æ\234\237å\200¤
               363:
               364:
                                                                                                           void update(int k, T val){ // kç\225°c\233®ã\201®å\200¤(0-indexed)ã\202\222
val ã\201«å¤\211æ\233´
                                                                                                                                                                   for (dat[k += n] = val; k > 0; k >= 1) { // k\tilde{a} 202 \ 222\ a \ 220 \ \alpha \ 202}
             365:
  \200a\214°\eq\226\223\ai\201\ai\ai\202\ai\203\\ai\203\203\\203\\203\\202\ai\202\\ai\202\\222\ai\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\ai\213\ai\201\\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201
 \202\211\(\ell\) \206\(\alpha\)\201\(\alpha\)\227\(\alpha\)\231
             366:
                                                                                                                                                                                                                            dat[k>>1] = func(dat[k], dat[k ^ 1]);
               367:
               368:
               369:
                                                                                                           T query(int 1, int r) { //[1, r)\tilde{a} \times 2010 \hat{a} \times 214 \hat{e} \times 226 \times 223
                370:
                                                                                                                                                                  T ret = neutral;
               371:
                                                                                                                                                                     for (1 += n, r += n; 1 < r; 1 >>= 1, r >>= 1)
               372:
                                                                                                                                                                                                                            if(1 & 1) ret = func(ret, dat[1++]);
               373:
                                                                                                                                                                                                                           if(r \& 1) ret = func(ret, dat[--r]);
```

```
374:
     375:
                                                        return ret;
     376:
     377: };
     378:
     379: int main(void){
     380:
                                    cin >> n >> d >> k;
     381:
                                    rep(i, n) cin >> x[i];
     382:
                                    segtree<pair<11, int>> seg(n, make_pair(0, 0));
     383:
                                    rep(i, n) seq.update(i, {x[i], -i}); //e^3/236 \times 234 \times 200 \times 234 \times 200 \times 217 
return 0;
     385: }
     386:
     387: /*
     388: http://vukicoder.me/problems/no/489
     389: (http://yukicoder.me/submissions/156443)
     390: */
     391:
     392:
     394: ##### ./data_struture/segtree_max_slow.hpp ######
     396:
     397: /*
     398: å\221¼ã\201³ã\201 ã\201\227æ\226¹
     399: segtree<int> seg(n); // intå\236\213ã\201®è|\201ç´ ã\201§ nã\201⁻è|\201ç´ ã\201®æ
     400: */
     401:
     402: template <class T> //T : dat[]ã\201@ā,-è°«ã\201@å\236\213
     403: class seqtree{
     404: public:
     405:
                                    int n;
     406:
                                    vector<T> dat;
     407:
                                     segtree(int n_): n(n_) \{ //n_e / 201c \approx 225^\circ \}
     408:
                                                        n = 1;
     409:
                                                        while (n < n) n *= 2;
     410:
                                                        dat.resize(n * 2, 0); //(1) å\210\235æ\234\237å\200\pi\202\222æ\234
\200a°\217ã\201« -INFã\201\213ã\202\202
     411:
     412:
                                    void update(int k, T val){ // kc\225 ac\233@ã\201@å\200¤(0-indexed)ã\202\222
val ã\201«å¤\211æ\2331
     413:
                                                        k += n - 1; //e\221\211a\201@c\200c\202^1
     414:
                                                        dat[k] = val;
     415:
                                                        while(k > 0)
     416:
                                                                           k = (k - 1) / 2;
     417:
                                                                           dat[k] = max(dat[k * 2 + 1], dat[k * 2 + 2]); // (2) å\214°é
\226\223ã\201@æ\234\200au§å\200uã\201§æ\233´æ\226°
     418:
     419:
                                    T query(int a, int b, int k, int l, int r){ //[a, b)\tilde{a}\201@æ\234\200a^s\tilde{a}\200a}
     420:
ã\202\222æ±\202ã\202\201ã\202\213
                                                        if(r <= a | b <= 1) return 0; //(3) æ\234\200aa§å\200¤ã\201«é\226¢ä
     421:
2 \sqrt{202} \sqrt{201} = \sqrt{201} = \sqrt{204} \sqrt{200} = \sqrt{201} = \sqrt{201} = \sqrt{202} = \sqrt{201} = \sqrt{2
     422:
                                                        if(a <= 1 && r <= b) return dat[k];</pre>
     423:
                                                        else{
     424:
                                                                           T vl = query(a, b, k * 2 + 1, 1, (1 + r) / 2);
     425:
                                                                           T vr = query(a, b, k * 2 + 2, (1 + r) / 2, r);
    426:
                                                                            return max(vl, vr); //(4) å\214°é\226\223ã\201@æ\234\200大å
\200¤ã\201§æ\233´æ\226°
    427:
     428:
     429:
                                    T query(int a, int b){
     430:
                                                        return query(a, b, 0, 0, n);
     431:
```

Page 5 of 17

HOGE University (team: HOGE) 432: };

```
433:
       434: /*
        435: segtree
        436: æ\233´æ\226°: 1 c\2021
       437: ã\202~ã\202~ã\203ª: å\214°é\226\223ã\201®æ\234\200大å\200¤
       438: note: (1)~(4)ã\202\222å¤\211ã\201\210ã\202\213ã\201\223ã\201"ã\201\§å*¾å;\234å\217-è
1203%
       439:
        440: http://codeforces.com/contest/777/problem/E
        441: (http://codeforces.com/contest/777/submission/25143272)
       443:
        444:
       446: ####### ./data struture/segtree min.hpp #########
        448:
       449: /*
        450: å\221¼ã\201³ã\201 ã\201\227æ\226¹
       451: segtree<int> seg(n); // intå\236\213ã\201@è|\201ç´ ã\201§ nã\201-è|\201ç´ ã\201@æ
12250
       452: */
       453:
       454: template <class T> //T : dat[]ã\201@ā,-è°«ã\201@å\236\213
        455: class segtree{
       456: public:
        457:
                                                         int n;
        458:
                                                         vector<T> dat;
        459:
                                                         segtree(int n_): n(n_){ //n_e/\201c \approx\225^o
        460:
                                                                                       n = 1;
        461:
                                                                                       while (n < n) n *= 2;
        462:
                                                                                        \200大ã\201«
        463:
        464:
                                                         void update(int k, T val){ // kc\225ac\233@ã\201@å\200¤(0-indexed)ã\202\222
val ã\201«å¤\211æ\2331
                                                                                       for (dat[k += n] = val; k > 0; k >>= 1) { // kã \ 202 \ 222å \ 220 «ã \ 202}
\200a\214°\eq\226\223\ai\201\ai\ai\202\ai\203\\ai\ai\203\207\ai\203\203\\203\\202\ai\202\\ai\202\\222\ai\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\
\202\211\(\ell\) \206\(\alpha\)\210\227\(\alpha\)\231
                                                                                                                      dat[k>>1] = min(dat[k], dat[k ^ 1]); // (2) å\214°é\226\223ã
\201@æ\234\200大å\200¤ã\201§æ\233´æ\226°
       467:
        468:
        469:
                                                        T query(int 1, int r) { //[1, r)\tilde{a} \times 2010 \hat{a} \times 214 \hat{e} \times 226 \times 223
       470:
                                                                                       T ret = INF; //(3) \approx 234 200 \approx 217 \approx 200 \approx 201 \approx 201
\201\204å\200¤
       471:
                                                                                        for (1 += n, r += n; 1 < r; 1 >>= 1, r >>= 1){
       472:
                                                                                                                     if(1 & 1) ret = min(ret, dat[1++]); //(4) a^{214} = (226) (223)
473:
                                                                                                                     if(r \& 1) ret = min(ret, dat[--r]); //(4) å\214°\elle\226\223\tilde{a}
\verb|\|201@x|234|200a^{\circ}|217a^{\circ}|200\|a||201\|x||233^{\circ}|x||226^{\circ}|
      474:
                                                                                      }
       475:
                                                                                       return ret;
       476:
        477: };
        478:
       479:
        480: /*
        481: segtree ã\201\231ã\201\223ã\201\227é«\230é\200\237å\214\226
        482: æ\233´æ\226°: 1 ç\202¹
       483: ã\202~ã\202~ã\203ª: å\214°\eq\226\223ã\201\@\234\200\å°\217\å\200\\\
        484:\ note:\ (1)^{\sim}(4)\tilde{a}\backslash202\backslash222\mathring{a}^{\alpha}\backslash211\tilde{a}\backslash201\backslash210\tilde{a}\backslash202\backslash213\tilde{a}\backslash201\backslash223\tilde{a}\backslash201^{\circ}\tilde{a}\backslash201\S\mathring{a}^{-}\mathring{A}\mathring{a};\backslash234\mathring{a}\backslash217^{-}\grave{e}
\2031/2
        485: http://arc045.contest.atcoder.jp/tasks/arc045_b
       486: (http://arc045.contest.atcoder.jp/submissions/1151765)
```

```
488: http://codeforces.com/contest/689/problem/D
    489: (http://codeforces.com/contest/689/submission/25354957)
    490: */
    491:
    492:
    494: ##### ./data struture/segtree min slow.hpp ######
    497: /*
    498: 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°
    500: */
    501:
    503: template <class T> //T : dat[]ã\201@ā,-è°«ã\201@å\236\213
    504: class segtree{
    505: public:
    506:
                                  int n:
    507:
                                  vector<T> dat;
    508:
                                  segtree(int n_): n(n_){ //n_e/\201ç^ x\225°
    509:
                                                     n = 1;
    510:
                                                     while(n < n_) n *= 2;</pre>
    511:
                                                     dat.resize(n * 2, INF); //(1) å\210\235æ\234\237å\200¤ã\202\222æ\234
\200大ã\201«
    512:
    513:
                                  void update(int k, T val){ // kc\225 ac\233@ã\201@å\200¤(0-indexed)ã\202\222
val ã\201«å¤\211æ\233´
    514:
                                                     k += n - 1; //e^{221}211a^{2010}c^{-200}c^{202}
    515:
                                                     dat[k] = val;
    516:
                                                     while(k > 0)
    517:
                                                                        k = (k - 1) / 2i
                                                                        dat[k] = min(dat[k * 2 + 1], dat[k * 2 + 2]); // (2) a\214°6
\226\223ã\201@æ\234\200å°\217å\200¤ã\201§æ\233´æ\226°
    519:
    520:
                                  T query(int a, int b, int k, int 1, int r) \{ //[a, b) \tilde{a} / 201 \Re (234) / 200 \Re (23
ã\202\222æ±\202ã\202\201ã\202\213
                                                     522:
\226¢ä;\202ã\201ªã\201\204å\200¤ã\201§æ\233´æ\226°
    523:
                                                     if(a <= 1 && r <= b) return dat[k];
    524:
                                                     else{
                                                                        T vl = query(a, b, k * 2 + 1, l, (l + r) / 2);
    525:
    526:
                                                                        T \text{ vr} = query(a, b, k * 2 + 2, (1 + r) / 2, r);
    527:
                                                                        return min(vl, vr); //(4) å\214°é\226\223ã\201@æ\234\200å°
\217å\200¤ã\201§æ\233´æ\226°
    528:
    529:
    530:
                                  T query(int a, int b){
    531:
                                                     return query(a, b, 0, 0, n);
    532:
    533: };
    534:
    535: /*
    536: segtree
    537: æ\233´æ\226°: 1 ç\2021
    538: \tilde{a} \setminus 202^{-}\tilde{a} \setminus 202^{-}\tilde{a} \setminus 203^{a}: \tilde{a} \setminus 214^{\circ} \in \setminus 226 \setminus 223\tilde{a} \setminus 201@a \setminus 234 \setminus 200\mathring{a}^{\circ} \setminus 217\mathring{a} \setminus 200 = 100
    539: note: (1)^{\sim}(4)\tilde{a}\setminus 202\setminus 222\dot{a} = 121\tilde{a}\setminus 201\setminus 210\tilde{a}\setminus 202\setminus 213\tilde{a}\setminus 201\setminus 223\tilde{a}\setminus 201\tilde{a} = 1201\tilde{a}\cdot 201\tilde{a}\setminus 201\tilde{a}\setminus 201\tilde{a}\setminus 201\tilde{a}\setminus 201\tilde{a}\setminus 201\tilde{a}\setminus 201\tilde{a}\setminus 201\tilde{a}
\2031/2
    541: http://arc045.contest.atcoder.jp/tasks/arc045_b
    542: (http://arc045.contest.atcoder.jp/submissions/1151282)
    543: */
    544:
```

```
547: ###### ./data_struture/segtree_monoid.cpp #######
  550: template <class T> //T(ã\203¢ã\203\216ã\202¤ã\203\211) : dat[]ã\201®å,-è°«ã\201®å
\236\213
  551: class segtree{
  552: public:
  553:
               int n;
  554:
               T neutral;
  555:
                vector<T> dat;
               T func(T 1, T r) { //\ddot{a}^{\circ} \setminus 214\acute{e} \setminus 205æ^{\prime}_{4} \setminus 224c^{\circ}_{2} \setminus 227\mathring{a} - \setminus 220
  556:
  557:
                        int lok, lopen, lclose, rok, ropen, rclose;
                        tie(lok, lopen, lclose) = 1;
  558:
  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) \left\{ \frac{201c}{w} \frac{225^{\circ} val}{val} \right\}
\230ä\215å\205\203
  564:
                        n = 1:
  565:
                        while(n < n_) n *= 2;</pre>
  566:
                        dat.resize(n * 2, neutral); //å\210\235æ\234\237å\200¤
  567:
  568:
               void update(int k, T val){ // kc\225°c\233®ã\201®å\200¤(0-indexed)ã\202\222
val ã\201«å¤\211æ\2331
                       k += n - 1;
  569:
  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 1, int r) { //[a, b)\tilde{a} \times 2018\dot{a} \times 214^{\circ}\dot{e} \times 226 \times 223
  577:
                        if(r <= a | | b <= 1) return neutral;</pre>
  578:
                        if(a <= 1 && r <= b) return dat[k];</pre>
  579:
                        else {
  580:
                                T vl = query(a, b, k * 2 + 1, 1, (1 + r) / 2);
  581:
                                T \text{ vr} = \text{query}(a, b, k * 2 + 2, (1 + 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:
  591: ##### ./data struture/segtree template.hpp ######
  593:
  594: template <class T> //T : dat[]ã\201@ā,-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 1, T r) { //\text{å} | 214^{\circ} \in (226 | 223\tilde{a} | 202 | 222\tilde{a} | 203 | 236\tilde{a} | 203 | 236\tilde{a} | 201 | 231\tilde{a}
return max(1, r); /* ã\201\223ã\201\223ã\202\222åa\211ã\201\210ã\202
  601:
\213 */
  602:
  603:
                segtree(int n_, T val): n(n_), neutral(val){ //n_:e/\201c^ a\225° val:a\215}
\230ä\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){ // kc\225ac\233@ã\201@å\200¤(0-indexed)ã\202\222
val ã\201«å¤\211æ\233°
                                         for (dat[k += n] = val; k > 0; k >>= 1) { // <math>k\tilde{a} \ 202 \ 222a \ 220 \ \tilde{a} \ 202}
\200a\214°\eq\226\223\ai\201\ai\202\ai\203\\ai\203\\207\ai\203\\203\\202\ai\202\\ai\202\\222\ai\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\201\213\ai\213\ai\213\ai\213\ai\213\ai\213\ai\213\ai\213\ai\213\ai\213\ai\213\a
\202\211\(\ell\) \206\(\alpha\)\210\227\(\alpha\)\231
                                                        dat[k>>1] = func(dat[k], dat[k ^ 1]);
   611:
   612:
   613:
                           T query(int 1, int r) { //[1, r)\tilde{a} \times 2010\hat{a} \times 214^{\circ}\hat{e} \times 226 \times 223
   614:
                                         /* å\217~æ\217\233å\207°ã\201°ã\201\204æ\231\202å\215±é\231° */
   615:
                                         T ret = neutral;
   616:
                                          for (1 += n, r += n; 1 < r; 1 >>= 1, r >>= 1)
   617:
                                                        if(1 \& 1) ret = func(ret, dat[1++1);
   618:
                                                        if(r \& 1) ret = func(ret, dat[--r]);
   619:
   620:
                                         return ret;
   621:
   622: };
   623:
   624: /*
   625: å\221¼ã\201³å\207°ã\201\227æ\226¹
   626: ä¾\213
   627: seqtree<pair<11, int>> seq(n, make pair(0, 0));
   629: funcã\201@é\203"å\210\206ã\202\222åp\211ã\201\210ã\201|ã\203\236ã\203\ã\202\ã\201
\231ã\202\213
   630:
   631: http://yukicoder.me/problems/no/489
   632: (http://yukicoder.me/submissions/156443)
   633: */
   634:
   636: ##### ./data_struture/SparseTable_max.hpp #######
   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@e|\201ç´ ã\201§ nã\201¯e|\201ç´ ã\201®
æ\225°
   643: */
   645: template <class T> //T : table[][]ã\201@ā,-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]\tilde{a}\201\tilde{a}\234\200\tilde{a}\$\tilde{a}\200\tilde{a}
   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:
                                          \226\223ã\201@æ\234\200ap$å\200p
   655:
                                                        table[i][0] = val[i];
   656:
   657:
                                          for (int k = 0; k < M - 1; ++k) \{ // [i, i + 2^{(k+1)}) \tilde{a} \ 201@ \tilde{a} \ 214^{\circ} \tilde{e} \}
\226\223ã\202\222è"\210ç@\227
                                                        for (int i = 0; i + (1<< k) < N; ++i){</pre>
   658:
                                                                       // iã\201\213ã\202\2112^(k+1)ã\201®é\225•ã\201\225ã
   659:
é\226\223ã\201@æ\234\200大å\200¤ã\202\222å\210@c\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\200大å\200¤
         661:
          662:
          663:
                                                                          T query(int 1, int r) \{ // O(1) [1, r) \tilde{a} \ 201 \% \ 226 \ 23 \tilde{a} \ 201 \% \ 234 \ 200 \% \ 234 \ 200 \% \ 236 \ 200 \% \ 236 \ 200 \% \ 236 \ 200 \% \ 236 \ 200 \% \ 236 \ 200 \% \ 236 \ 200 \% \ 236 \ 200 \% \ 236 \ 200 \% \ 236 \ 200 \% \ 236 \ 200 \% \ 236 \ 200 \% \ 236 \ 200 \% \ 236 \ 200 \% \ 236 \ 200 \% \ 236 \ 200 \% \ 236 \ 200 \% \ 236 \ 200 \% \ 236 \ 200 \% \ 236 \ 200 \% \ 236 \ 200 \% \ 236 \ 200 \% \ 236 \ 200 \% \ 236 \ 200 \% \ 236 \ 200 \% \ 236 \ 236 \ 236 \ 200 \% \ 236 \ 236 \ 236 \ 200 \% \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 236 \ 23
          664:
                                                                                                                  int k = 31 - builtin clz(r - 1); //å \langle 214°6 \rangle \langle 223 \tilde{a} \rangle \langle 201@6 \rangle \langle 225°\tilde{a} \rangle
           665:
201 225\tilde{a} 201 \tilde{a} 215 212 \tilde{a} 210 206 \tilde{a} \tilde{a} \tilde{a} 210 \tilde{a} 201 \tilde{a} 200 \tilde{a} 200 \tilde{a} 200 \tilde{a} 200 \tilde{a}
                                                                                                                  return max(table[1][k], table[r - (1 << k)][k]); // (2) æ\234\200au8
å\200¤
          667:
            668: };
           669:
           670:
           671: /*
           672: å\211\215å\207/c\220\206(nlogn)
          673: æ\234\200å°\217å\200¤(1)
          674:
           675: http://codeforces.com/contest/689/problem/D
           676: (http://codeforces.com/contest/689/submission/25363691)
           677: */
           678:
          679:
          680:
           682: ##### ./data_struture/SparseTable_min.hpp #######
          684:
          685: /*
          686: å\221¼ã\201³ã\201 ã\201\227æ\226¹
          687: int v[1000];
           688: seqtree<int> seq(n, v); // intå\236\213ã\201@è|\201c´ã\201s nã\201-è|\201c´ã\2010
æ\225°
           689: */
           690:
           691: template <class T> //T : table[][]ã\201@ā,-è°«ã\201@å\236\213
          692: class SparseTable{
           693: public:
          694:
                                                                           int N, M; //table[N][M]
           695:
                                                                           // \text{ table[i][k]} := [i, i + 2^k)\tilde{a} \ 2010 \times 234 \ 200 \circ \ 217 \circ \ 200 
           696:
                                                                           vector<vector<T>> table;
           697:
                                                                           template < class S > SparseTable(int n, S &val): N(n) { // O(nlogn)
                                                                                                                 M = 32 - builtin clz(N); // M - 1 <= logN < M
           698:
           699:
                                                                                                                   table.resize(N. vector<T>(M));
                                                                                                                  for (int i = 0; i < N; ++i) { // [i, i + 1) \( \tilde{a} \) \( 201 \) \( 201 \) \( 201 \) \( 201 \) \( 201 \) \( 201 \) \( 201 \) \( 201 \) \( 201 \) \( 201 \) \( 201 \) \( 201 \) \( 201 \) \( 201 \) \( 201 \) \( 201 \) \( 201 \) \( 201 \) \( 201 \) \( 201 \) \( 201 \) \( 201 \) \( 201 \) \( 201 \) \( 201 \) \( 201 \) \( 201 \) \( 201 \) \( 201 \) \( 201 \) \( 201 \) \( 201 \) \( 201 \) \( 201 \) \( 201 \) \( 201 \) \( 201 \) \( 201 \) \( 201 \) \( 201 \)
  \226\223ã\201@æ\234\200å°\217å\200¤
          701:
                                                                                                                                                         table[i][0] = val[i];
          702:
          703:
                                                                                                                  for (int k = 0; k < M - 1; ++k) \{ // [i, i + 2^{(k+1)}) \tilde{a} \setminus 201 \hat{a}^{(k+1)} \}
 \226\223ã\202\222è"\210¢®\227
         704:
                                                                                                                                                         for (int i = 0; i + (1 << k) < N; ++i){
          705:
                                                                                                                                                                                                 // iã\201\213ã\202\2112^(k+1)ã\201®é\225•ã\201\225ã
 6 \ 226 \ 223 \tilde{a} \ 201 \ \tilde{a} \ 234 \ 200 \ \tilde{a} \ 217 \ \tilde{a} \ 200 \ \tilde{a} \ 222 \ \tilde{a} \ 210 \ \tilde{c} \ \sqrt{224} \ \tilde{a} \ 201 \ \sqrt{227} \ \tilde{a} \ 201 \ / \ \tilde{a} \ 202 \ / \ 202 \ \tilde{a} \ 202 \ / \ 202 \ \tilde{a} \ / \ 202 \ / \ 202 \ \tilde{a} \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ / \ 202 \ 
\213
         706:
                                                                                                                                                                                                  table[i][k + 1] = min(table[i][k], table[i + (1 << k])
)][k]); // (1)æ\234\200å°\217å\200¤
         707:
          708:
          709:
          710:
                                                                         T query(int 1, int r) \{ // O(1) [1, r) \tilde{a} | 201 \tilde{e} | 226 | 223 \tilde{a} | 201 \tilde{e} | 234 | 200 \tilde{a}^{\circ} | 217 \tilde{a} | 201 \tilde{e} | 218 \tilde{a} | 201 \tilde{e} | 218 \tilde{a} | 21
\200¤
                                                                                                                   int k = 31 - __builtin_clz(r - 1); //å \ 214°6 \ 226 \ 223ã \ 201@6 \ 225•ã
\201\225ã\201@å\215\212å\210\206以ä,\212ã\201@å\200¤ ï¼\210k<= r - 1 < k + 1)
         712:
                                                                                                                   return min(table[1][k], table[r - (1 << k)][k]); // (2) æ\234\200å°
```

```
\217å\200g
                  713:
                  714: };
                 715:
                  716: /*
                  717: template \langle class\ T \rangle //T: table[][]\tilde{a} \langle 201@\tilde{a}, -\hat{e}^{\circ} \langle \tilde{a} \rangle \langle 201@\tilde{a} \rangle \langle 236 \rangle \langle 213 \rangle 
                  718: class SparseTable{
                  719: public:
                  720:
                                                                                                        int N, M; //table[N][M]
                  721:
                                                                                                        // table[i][k] := [i, i + 2^k) \( \tilde{a} \) \( 234 \) \( 200 \tilde{a} \) \( 217 \tilde{a} \) \( 200 \tilde{a} \)
                  722:
                                                                                                         vector<vector<T>> table;
                  723:
                                                                                                         template < class S > SparseTable(int n, S &val): N(n) { // O(nloqn)
                  724:
                                                                                                                                                               M = 32 - builtin clz(N); // M - 1 <= logN < M
                  725:
                                                                                                                                                               table = vector < T > (N, vector < T > (M, INF)); // (1) a \ 234 \ 200 a^{\circ}
    \217\(\alpha\)\201\(\alpha\)\226\(\alpha\)\\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\(\alpha\)\201\
                                                                                                                                                               for (int i = 0; i < N; ++i) { // [i, i + 1)ã\201¾ã\201§ã\201®å\214°é
    \226\223ã\201@æ\234\200å°\217å\200¤
                 727:
                                                                                                                                                                                                                     table[i][0] = val[i];
                  728:
                  729:
                                                                                                                                                                for (int k = 1; k < M; ++k){ // [i, i + 2^k)ã\201@å\214°é\226\223ã
    \202\222e"\210c@\227
                 730:
                                                                                                                                                                                                                     for (int i = 0; i + (1 << k) <= N; ++i)
                                                                                                                                                                                                                                                                           // iã\201\213ã\202\2112^kã\201®é\225•ã\201\225ã\201®
                 731:
   å\214°é\226\223ã\201@æ\234\200å°\217å\200¤ã\202\2222^(k-1)ã\201@é\225•ã\201\225ã\201@å\214°é
    \226\223ã\201@æ\234\200å°\217å\200¤ã\202\222å\210@c\224"ã\201\227ã\201\x±\202ã\202\201ã\202
    1213
               732:
                                                                                                                                                                                                                                                                            auto first = table[i][k - 1];
                 733:
                                                                                                                                                                                                                                                                            auto second = table[i + (1 << (k - 1))][k - 1];
                                                                                                                                                                                                                                                                            table[i][k] = (first < second) ? first : second; //</pre>
    (2) å°\217ã\201\225ã\201\204æ\226¹ã\201§æ\233´æ\226°
                 735:
                 736:
                  737:
                  738:
                                                                                                        T query(int 1, int r){ // O(1) [1, r) \tilde{a} \geq 0.0 = 2.23 = 2.01 = 2.34 \geq 0.04 = 2.01 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 = 0.04 
    \200¤
               739:
                                                                                                                                                                int k = 31 - builtin clz(r - 1); //a \langle 214^{\circ} \in \langle 226 \backslash 223a \langle 201 \otimes e \rangle \langle 225^{\bullet} a \rangle \langle 214^{\circ} \in \langle 214 \rangle \langle 214^{\circ} \otimes \langle 214^{\circ} \in \langle 214 \rangle \langle 214^{\circ} \otimes 
    740:
                                                                                                                                                               return min(table[1][k], table[r - (1 << k)][k]); // (3) <math>x^2
    \217å\200g
                  741:
                 742: };
                  743: */
                  744:
                  745:
                  747: å\211\215å\207/c\220\206(nlogn)
                  748: æ\234\200å°\217å\200¤(1)
                  750: http://arc045.contest.atcoder.jp/tasks/arc045_b
                  751: (http://arc045.contest.atcoder.jp/submissions/1152995)
                  752: (http://arc045.contest.atcoder.jp/submissions/1152999)
                  \206\alpha\201\alpha\201\231\alpha\202\213\alpha\201\213
                 754: */
                 755:
                  756:
                  759: ######## ./data_struture/union_find.h #########
                  761:
                  762: class DisjointSet{
                  763: public:
                 764:
                                                                                                         vector<int> rank, p;//rank:æ\234~ã\201®é«\230ã\201\225 p:è|ªã\201®é\202ç
    \2021c\225aå\217.
                 765:
                                                                                                         DisjointSet(){}
```

```
766:
                                           DisjointSet(int size){//é \202c\2021ã\2010æ\2250
     767:
                                                                  rank.resize(size, 0);
     768:
                                                                  p.resize(size, 0);
     769:
                                                                  rep(i, size) makeSet(i);
     770:
     771:
                                           bool same(int x, int y){ //å\220\214ã\201\230æ\234~ã\201\201\202ã\202\213ã
 \201\213
                                                                   return findSet(x) == findSet(y);
     772:
     773:
     774:
                                           void unite(int x, int v){ // æ\234"ã\201@ã\201\206ã\201\227ã\202\222ã\201
 \217\(\alpha\) 201\(\alpha\) 201\\221\(\alpha\) 202\\213
     775:
                                                                 link(findSet(x), findSet(v));
      776:
      777: private:
      778:
                                           void makeSet(int x){
                                                                  p[x] = x;
      779:
      780:
                                                                  rank[x] = 0;
      781:
     782:
                                           int findSet(int x){ //è/ªã\202\222æ\216¢ã\201\231ï\210ã\203\8ã\203\8ã\203\210
ã\201%ã\2018ï\211
     783:
                                                                  if(x = |x|)
                                                                                        p[x] = findSet(p[x]);
      784:
      785:
      786:
                                                                  return p[x];
     787:
     788:
                                           void link(int x, int y){ //æ\234~ã\201@é«\230ã\201\225ã\202\222è\200\203æ
\205@\alpha\201\227\alpha\201\\alpha\234\\alpha\201\206\alpha\201\227\alpha\202\222\alpha\201\217\alpha\201\\alpha\21\alpha\201\221\alpha\201\217\alpha\201\\alpha\21\alpha\201\221\alpha\201\\alpha\21\alpha\201\\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21\alpha\21
\202\213
     789:
                                                                  if(rank[x] > rank[y]){
     790:
                                                                                         p[y] = x;
     791:
                                                                   }else{
      792:
                                                                                         p[x] = y;
      793:
                                                                                         if(rank[x] == rank[y]) rank[y]++;
      794:
      795:
      796: };
      797:
     798: /*
     799: http://abc049.contest.atcoder.jp/tasks/arc065 b
     801:
      805:
      807: * ä°\214æ¬;å\205\203c´¬c©\215å\222\214(é\225•æ\226¹å¾¢å\206\205ã\201®å\200¤ã\201®å
\220\210\eartitle{c}"\210\eartitle{a}\202\222\eartitle{a}\207\eartitle{a}\201\231\eartitle{a}\225\217\eartitle{e};\214\)
     808: */
     809:
     810: // O(hw)
     811: void sum2D(int h, int w) { // a ã\201@c./ã\201"æ"aã\201@é\225•ã\201\225
     812:
                                           rep(y, h + 1)rep(x, w + 1) sum[y][x] = 0;
     813:
                                           \237\213ã\202\201è¾¼ã\202\200
     814:
                                           rep(y, h + 1)rep(x, w) sum[y][x + 1] += sum[y][x]; // æ^a
     815:
                                           rep(y, h)rep(x, w + 1) sum[y + 1][x] += sum[y][x]; // c./
     816: }
     817: // 0(1)
     818: int calcsum(int y1, int x1, int y2, int x2) { // æ±\202\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 201\\\ 20
\225•æ\226¹å½¢ã\201®å•|ä,\212, å\217³ä,\213ã\201®å°§æ"\231
     819:
                                          return sum[y2 + 1][x2 + 1] - sum[y2 + 1][x1] - sum[y1][x2 + 1] + sum[y1][x1]
     820: }
     821:
     822: /*
```

```
823: "ã\201\43\201\232ã\201<sup>-</sup>å\237\213ã\202\201è\\43\202\200" ã\201@\203"å\210\206ã\201@aã
824: aoi 1176
 825:
 826: */
 827:
 829: ########## ./dp/Traveling Salesman.h ############
 832: const int MAX N = 20;
 833: vector<pair<int. int> > G[MAX N]; /* G[u][v] := u->v's weight */
 834: int dp[(1 << MAX N)][MAX N];
 835: int pre[(1 << MAX N)][MAX N];
 836: int Traveling Salesman(int n) { /* n := Number of vertices */
             rep(i, (1 << n))rep(i, n)dp[i][i] = INF; /* Initialization */
 838:
             0 = [0][0]ab
 839:
             for (int mask = 0; mask < (1 << n); ++mask){</pre>
 840:
                    for (int u = 0; u < n; ++u){ /* Current vertex */</pre>
                           for(auto p : G[u]){ /* Next vertex */
 841:
 842:
                                   int v = p.first, w = p.second;
 843:
                                   if((mask & (1 << v)) == 0){
 844:
                                          if(dp[mask | (1 << v)][v] > dp[mask][u] + w)
 845:
                                                 dp[mask | (1 << v)][v] = dp[mask][u]
+ w;
 846:
                                                 pre[mask | (1 << v)][v] = u;
 847:
 848:
 849:
 850:
 851:
 852:
             return dp[(1 << n) - 1][0];
 853: }
 854: vector<int> Restoration(int n){
 855:
             vector<int> root;
 856:
             int mask = (1 << n) - 1, v = 0;
 857:
             root.push back(v);
 858:
             while(mask != 0){
 859:
                    int u = pre[mask][v]; /* u -> v */
 860:
                    root.push back(u);
 861:
                    mask ^= (1 << v);
 862:
                    v = u;
 863:
 864:
             reverse(root.begin(), root.end());
 865:
             return root;
 866: }
 867:
 869: http://judge.u-aizu.ac.jp/onlinejudge/description.jsp?id=DPL_2_A
 870: cu\214è•-åi@å\205\203ã\201-æq\234è"½ã\201\227ã\201\ã\201\204ã\201\204
 871: */
 872:
 874: ########## ./geometry/template.cpp ############
 876:
 877: class Point { // c\2021
 878: public:
 879:
         double x, y;
 880:
         Point(double x = 0, double y = 0):x(x), y(y){}
 881:
         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); }
 882:
         Point operator * (double a) const { return Point(a * x, a * y); }
 883:
         Point operator / (double a) const { return Point(x / a, y / a);
 884:
 885:
         double abs() const { return sqrt(norm()); }
```

```
double norm() const { return x * x + v * v; }
     887:
                               // bool operator < (const Point &p) const { return x != p.x ? x < p.x : y < p.y;
                                888:
æ \ 224è¼\ 203
                                          return x + EPS < p.x || (eq<double>(x, p.x) && y + EPS < p.y);</pre>
     889:
      890:
      891:
                               bool operator == (const Point &p) const { return (eq<double>(x, p.x) && eq<doubl
e>(y, p.y)); }
      892: };
      893: using Vector = Point;
      894: using Polygon = vector<Point>; // åp\232è\222å\c/c
      896: double dot(const Vector& a, const Vector& b) { return a.x * b.x + a.y * b.y; } // ã
897: double cross(const Vector& a, const Vector& b) { return a.x * b.v - a.v * b.x; } //
ã\203\231ã\202~ã\203\210ã\203\aaa\201"bã\201@åg\226c@\215
      898: double length2(const Point& a) { return a.norm(); } // é\200\232å, ã\201@é\225•ã\201
\225ã\201@2ä1\227
     899: double length(const Point& a) { return a.abs(); } // é\200\232å, ~~a\201@é\225.~a\201
     900: Point rotationalTransfer(Point c, double r, double deg) { // cã\202\22ä,-å¿\203ã
§æ"\231
     901:
                               double rad = PI * deg / 180.0; return c + Point(cos(rad), sin(rad)) * r;
     902: }
     903: // (x, y, z) \tilde{a} 201@c 202^{1}\tilde{a} 202 222^{4} 205 211æ^{200} (xy^{4}^{8} x^{2} 231\tilde{a} 201\tilde{a} 2018^{2} 222^{4})
\201\214thetaå°|, xyå¹³é\235¢ã\201\213ã\202\211zæ\226¹å\220\221ã\201.ã\201®è§\222å°|ã\201
ã\201@å½±ã\201@xyå°§æ"\231
      904: Point Shadow(double x, double y, double z, double theta, double phi) {
                               theta = PI * theta / 180.0, phi = PI * phi / 180.0;
      906:
                               return Point(x - z / tan(phi) * cos(theta), y - z / tan(phi) * sin(theta));
      907: }
      908:
      909: enum ccw t {
                               COUNTER_CLOCKWISE = 1, // p0 \rightarrow p1 a^217 215x^21^202e^2 210a^23^26a^202^212a^201
@æ\2261å\220\221ã\201«p2
     911:
                               CLOCKWISE = -1, // p0 \rightarrow p1 \approx 231 \times 202 \approx 2104 \times 233 \times 236 \times 202 \times 2126 \times 2018 \approx 226 \approx 2018 \approx 220 \times 2018 \approx 2018 \approx
ã\201«p2
      912:
                                ONLINE BACK = 2, // p2->p0->p1 ã\201@é\206ã\201§c\233´c•\232ä\\212ã\201§p2
     913:
                                ONLINE FRONT = -2, // p0->p1->p2 \tilde{a} \setminus 20186 \setminus 206\tilde{a} \setminus 2018c \setminus 233^{\circ}c \cdot \setminus 232\tilde{a} \setminus 212p2
      914:
                                915: };
      916: ccw_t ccw(Point p0, Point p1, Point p2) {
                              Vector a = p1 - p0, b = p2 - p0;
      918:
                               if ( cross(a, b) > EPS ) return COUNTER_CLOCKWISE;
                              if ( cross(a, b) < -EPS ) return CLOCKWISE;</pre>
                               921:
                               if ( a.norm() < b.norm() ) return ONLINE FRONT;</pre>
      922:
                               return ON SEGMENT;
      923: }
      924:
      925: class Segment { //c•\232å\210\206
      926: public:
      927:
                               Point p1, p2;
      928:
                               Segment(){}
      929:
                                Segment(Point p1, Point p2):p1(p1), p2(p2){}
      930: };
      931: using Line = Segment;
      932:
      933: // *** å¤\232è§\222å½¢ ***
      934: // IN := 2, ON := 1, OUT := 0
      935: vector<Segment> getPolygonSegument(const Polygon& p) { //åa\232è§\222å½¢ã\201@c\2021
\tilde{a} \ 201 \ 213\tilde{a} \ 202 \ 211\tilde{a} \ 232\tilde{e} \ 222\tilde{a} \ 26 \ 201 \tilde{e} \ 202 \ 222\tilde{a} \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202 \ 202
```

```
936:
          vector<Segment> ret;
 937:
         rep(i, p.size() - 1) ret.push_back(Segment(p[i], p[i + 1]));
 938:
         ret.push_back(Segment(p[p.size() - 1], p[0]));
 939:
         return ret;
 940: }
 202^{1}pã201214å220«ã201½ã202214ã201/ã201204ã202213ã201213
         int n = q.size(); bool x = false;
 943:
         for (int i = 0; i < n; ++i) {</pre>
 944:
             Point a = q[i] - p, b = q[(i + 1) % n] - p;
 945:
             if (abs(cross(a, b)) < EPS && dot(a, b) < EPS ) return 1;</pre>
 946:
             if (a.v > b.v) swap(a, b);
 947:
             if (a.v < EPS && EPS < b.v && cross(a, b) > EPS ) x = !x;
 948:
 949:
         return (x ? 2 : 0);
 950: }
 951: Polygon andrewScan(Polygon s) { // å\207 å\214\205(æ\234\200ã\202\202å•/ã\201«ã\201
Polygon u, 1;
 952:
 953:
         if (s.size() < 3) return s;</pre>
 954:
          sort(s.begin(), s.end()); // x, yã\202\222å\237°æ°\226ã\201«æ\230\2076\206ã\202
%ã\203¼ã\203\210
 955:
         // xã\201\214å°\217ã\201\225ã\201\204ã\202\202ã\201®ã\201\213ã\202\2112ã\201¤ u
ã\201«è;½å\212
 956:
         u.push_back(s[0]), u.push_back(s[1]);
 957:
         // x ã\201\214大ã\201\215ã\201\204ã\202\202ã\201®ã\201\213ã\202\2112ã\201¤1ã
\201«è¿½å\212
 958:
         1.push_back(s[s.size() - 1]), 1.push_back(s[s.size() - 2]);
 959:
         // å\207.å\214\205ã\201®ä.\212\203"ã\202\222\224\237æ\210\220
 960:
         for (int i = 2; i < s.size(); i++) {</pre>
 961:
             for (int n = u.size(); n \ge 2 && ccw(u[n - 2], u[n - 1], s[i]) != CLOCKWISE;
n--){
 962:
                 u.pop back();
 963:
 964:
             u.push back(s[i]);
 965:
 966:
         // å\207,å\214\205ã\201®ä,\213\\203"ã\202\222\224\237\\210\220
 967:
         for (int i = s.size() - 3; i >= 0; i--) {
 968:
             for (int n = 1.size(); n >= 2 && ccw(1[n - 2], 1[n - 1], s[i]) != CLOCKWISE;
n--){
 969:
                 1.pop back();
 970:
 971:
             1.push back(s[i]);
 972:
 973:
         // æ\231\202è"\210å\233\236ã\202\212ã\201«ã\201°ã\202\213ã\202\210ã\201\206ã\201
«å\207 å\214\205ã\201@c\202¹ã\201@å\210\227ã\202\222c\224\237æ\210\220
         reverse(l.begin(), l.end());
 975:
         for (int i = u.size() - 2; i >= 1; i--) l.push_back(u[i]);
 976:
         return 1;
 977: }
 978:
 979:
 980: // *** c•\232å\210\206ã\201@ä°¤å•@å\210¤å@\232 ***
 981: bool intersect(const Point& p1, const Point& p2, const Point& p3, const Point& p4) {
 982:
         return ( ccw(p1, p2, p3) * ccw(p1, p2, p4) <= 0 &&
 983:
                  ccw(p3, p4, p1) * ccw(p3, p4, p2) <= 0);
 984: }
 985: bool intersect(const Segment& s1, const Segment& s2) { // 交å•@ã\201\227ã\201\ã\201
\204ã\201\237ã\202\211true
 986:
         return intersect(s1.p1, s1.p2, s2.p1, s2.p2);
 987: }
 988: //*** c.\232å\210\206ã\201@ä°¤c\2021 ***
 989: Point getCrossPoint(Segment s1, Segment s2) { // c•\232å\210\206ã\201@ä°¤ç\2021ã\201
\201\206\\201\223\\201"
         Vector base = s2.p2 - s2.p1;
```

Page 10 of 17

```
991:
                                  double d1 = abs(cross(base, s1.p1 - s2.p1)), d2 = abs(cross(base, s1.p2 - s2.p1))
);
      992:
                                  double t = d1 / (d1 + d2);
       993:
                                  return s1.p1 + (s1.p2 - s1.p1) * t;
       994: }
      995: // *** è•\235é\233¢ ***
       996: double getDistance(Point& a, Point& b) { // c\2021aã\201"c\2021bã\201@e.\235\233¢
      997:
                                  return length(a - b);
      998:
      999: double getDistanceLP(Line& 1, Point& p) { // c\233´c•\232sã\201"c\201@è•\235é
 \233¢
   1000:
                                  return length(cross(1.p2 - 1.p1, p - 1.p1) / length(1.p2 - 1.p1));
   1001: }
   1002: double getDistanceSP(Segment s, Point p) { // c • \232å\210\206sã\201"c\2021pã\201®è •
 \235é\233¢
   1003:
                                  if( dot(s.p2 - s.p1, p - s.p1) < EPS ) return length(p - s.p1);</pre>
   1004:
                                  if( dot(s.p1 - s.p2, p - s.p2) < EPS ) return length(p - s.p2);</pre>
   1005:
                                  return getDistanceLP(s, p);
   1006: }
   1007: double getDistanceSS(const Segment& s1, const Segment& s2) { // ç•\232å\210\206s1ã
 \201"c•\232å\210\206s2ã\201@ä°¤c\2021
   1008:
                                  if( intersect(s1, s2) ) return 0.0; //ä°¤ã\202\217ã\201fã\201\204ã\202\213
ã\201"ã\201\215
   1009:
                                 return min(min(getDistanceSP(s1, s2.p1), getDistanceSP(s1, s2.p2)),
   1010:
                                                                   min(getDistanceSP(s2, s1.p1), getDistanceSP(s2, s1.p2)));
   1011: }
   1012: double getDistancePolP(const Polygon& pol, const Point& p) { // å¤\232è§\222å½¢polã
\201"c\2021pã\201@è•\235é\233¢
                                  if(contains(pol, p) != 0) return 0.0; // c\2021ã\201\214å¤\232è§\222å½¢ã\201@å
   1013:
1014:
                                  double ret = 1e9;
   1015:
                                  for(Segment& u : getPolygonSegument(pol)) ret = min(ret, getDistanceSP(u, p));
   1016:
   1017: }
   1018: double getDistancePolPol(const Polygon& p1, const Polygon& p2) { // å¤\232è§\222å¾¢p
                                  for(const Point& p : p1) if(contains(p2, p) != 0) return 0.0; // p1ã\201@c\2021ã
\201\214an\232è\222a\perp2\angle p2\angle 201\@a.-\angle 201\end{a}\200\\\angle a\201\angle 201\angle a\201\angle 201\angle a\201\angle 201\angle a\201\angle 201\angle a\201\angle 201\angle a\201\angle 201\angle a\201\angle a\201\angl
                                  for(const Point& p : p2) if(contains(p1, p) != 0) return 0.0; // p2ã\201@ç\2021ã
\201\214å\222\8\222\8\p1\ai\201\\ai\201\\ai\202\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\201\\ai\20
   1021:
                                  double ret = 1e9;
   1022:
                                  for(const Segment& u : getPolygonSegument(p1))for(const Segment& v : getPolygonS
egument(p2)) {
   1023:
                                              ret = min(ret, getDistanceSS(u, v));
   1024:
   1025:
                                  return ret;
   1026: }
   1027:
   1029: class Rectangle { // é\225•æ\226¹å½¢
   1030: public:
                                 // 3 2
   1031:
   1032:
                                  // 0 1 (å\217\215æ\231\202è"\210å\233\236ã\202\212ã\201«é\225•æ\226¹å½¢ã\201®é
 1033:
                                  vector<Point> p; // c\2021ã\202\222é \206c\225ªã\201«ã\201\204ã\202\214ã\202\213
ã\201\223ã\201"
   1034:
                                  Rectangle(vector<Point>&p):p(p) {
                                              rep(i, 3) reps(j, i + 1, 4) { //é\201@å½\223ã\201ªé\206ç\225ªã\201«ã\201
   1035:
 \204\alpha\202\214\alpha\201|\alpha\202\202\alpha\alpha\alpha\201\alpha\alpha\alpha\201\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\al
   1036:
   1037:
                                                           rep(k, 4) if(k != i \&\& k != j) {
   1038:
                                                                       cnt += ccw(p[i], p[j], p[k]) == COUNTER_CLOCKWISE;
   1039:
   1040:
                                                           if(cnt == 2) {
   1041:
                                                                       swap(p[i + 1], p[j]);
   1042:
                                                                       break;
```

```
1043:
 1044:
 1045:
 1046:
                   bool intersect(const Segment& s) { // c • \232å\210\206sã\201"é\225 • æ\226¹å½¢ã\201
@å°\221ã\201ªã\201\217ã\201"ã\202\2021è¾°ã\201\214䰤啮ã\201\227ã\201\ã\201\204ã\202\214ã
\201°true
 1047:
                          bool flag = false;
 1048:
                          rep(i, 4) flag |= ::intersect(s, Segment(p[i], p[(i + 1) % 4]));
 1049:
                          return flag;
 1050:
 1051:
                   bool contain(const Point& pp) { // c\2021ppã\201\214\225.æ\2261\&\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\chick\
\201«å\220«ã\201¾ã\202\214ã\202\214ã\201°(è¾°ã\202\222å\220«ã\201¾ã\201ªã\201\204)true
 1052:
                          bool flag = true;
 1053:
                          rep(i, 4) flag &= ccw(p[i], p[(i + 1) % 4], pp) == COUNTER CLOCKWISE;
 1054:
                          return flag;
 1055:
 1056:
                   bool contain(const Segment& s) { // c • \232å\210\206sã\201\214\6\225 • \alpha\2261å\6\cdot \alpha
rue
 1057:
                          return contain(s.p1) && contain(s.p2);
 1058:
 1059: };
 1060:
 1061:
 1062: class Circle {
 1063: public:
 1064:
                          Point c;
 1065:
                          double r;
 1066:
                          Circle(Point c = Point(), double r = 0.0):c(c), r(r) {}
 1067: }
 1068: double arg(Vector p) { return atan2(p.y, p.x); }
 1069: Vector polar(double a, double b) { return Point(cos(r) * a, sin(r) * a); }
 1070: pair<Point, Point> getCrossPoints(Circle c1, Circle c2) {
 1071:
                          assert(intersect(c1, c2));
 1072:
                          double b = abs(c1.c - c2.c);
 1073:
                          double a = acos(c1.r * c1.r + d * d - c2.r * c2.r) / (2 * c1.r * d);
 1074:
                          double t = arg(c2.c - c1.c);
 1075:
                          return make pair(c1.c + polar(c1.r, t + a), c1.c + polar(c1.r, t - a));
 1076: }
 1077:
 1078: /*
 1079: http://judge.u-aizu.ac.jp/onlinejudge/description.jsp?id=2009
 1080: http://judge.u-aizu.ac.jp/onlinejudge/description.jsp?id=1157&lang=jp
 1081: http://judge.u-aizu.ac.jp/onlinejudge/description.jsp?id=2402
 1082: */
 1083:
 1085: ######## ./geometry/types_triangles.h #########
 1087:
 1088: class Types_triangles{
 1089: public:
 1090:
                          int n;
 1091:
                          vector<int> y, x;//yå°§æ"\231ã\200\200xå°§æ"\231
                          const double EPS = 1e-10;
 1092:
 1093:
                          long long cnt_chokaku = 0, cnt_donkaku = 0, cnt_eikaku = 0;
 1094:
                          Types_triangles(const vector<int> &ty, const vector<int> &tx, int size):y(ty
), x(tx), n(size) { //n := å §x" \ 231ã \ 201@x \ 225°}
 1095:
                                        count();
 1096:
 1097: private:
 1098:
                          void count(){
 1099:
                                         for (int i = 0; i < n; ++i){</pre>
                                                       //-M_PI ~ M_PI(-180 ~ 180)
 1100:
 1101:
                                                       vector<double> angle;//(x[i], y[i])\tilde{a} \langle 202 \langle 222a^{2} \langle 216 \rangle \langle 237c \rangle \langle 202^{1}\tilde{a} \rangle
\201"ã\201\227ã\201\237å\201\217è§\222
```

Page 11 of 17

```
for (int j = 0; j < n; ++j){
    1103:
                                                                                                                                          if(j == i) continue;
    1104:
                                                                                                                                           angle.push_back(atan2(y[j] - y[i], x[j] - x[i]);
    1105:
    1106:
                                                                                                               sort(angle.begin(), angle.end());
    1107:
                                                                                                               for (int j = 0; j < n - 1; ++j){//2\text{å}/221\text{°c}/233\text{@ã}/202/2223\%}
 \234ã\202\213
    1108:
                                                                                                                                           angle.push back(angle[j] + M PI*2);
    1109:
    1110:
                                                                                                               for (int j = 0; j < n - 1; ++j){
                                                                                                                                           cnt chokaku += upper bound(angle.begin(), angle.end(
    1111:
), angle[i] + M PI/2.0 + EPS) - lower bound(angle.begin(), angle.end(), angle[i] + M PI/2.0
- EPS);
   1112:
                                                                                                                                           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);
   1113:
    1114:
    1115:
                                                                                  cnt_eikaku = (11)n * (n - 1) * (n - 2) / 6 - cnt_chokaku - cnt_donka
ku;
   1116:
    1117: };
    1118:
    1119: /*
    1120: \ \ d^{\circ}\$e^{-1} / 2318 / 201 / 2148 / 201 / 2028 / 201 / 2378 / 201 / 2108 / 202 / 2118 / 202 / 2148 / 201 / 3(200 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 2018 / 20
1121: é\213-è§\222ã\200\201é\210\215è§\222ã\200\201¢\233´è§\222ã\201\214ã\201\235ã\202\214
\tilde{a} \setminus 201 \setminus 236\tilde{a} \setminus 202 \setminus 214\tilde{a} \setminus 201 \setminus 204\tilde{a} \setminus 201 \setminus 217\tilde{a} \setminus 201 \mu \tilde{a} \setminus 201 \setminus 232\tilde{a} \setminus 201 \mu \tilde{a} \setminus 201 \setminus 215\tilde{a} \setminus 202 \setminus 213\tilde{a} \setminus 201 \setminus 213\tilde{a} \setminus 213\tilde{
\202\222\alpha\202\201\alpha\202\213\alpha\200\202
   1122: http://abc033.contest.atcoder.jp/tasks/abc033_d
    1123: */
    1124:
    1129: const int MAX_N = 210;
    1130: using TYPE = double; // è · \235\eqrig \235\eqrig \201\@a\236\213\202\222\222\205\eqrig \202\214\202\213
    1131: vector<pair<int, TYPE> > G[MAX N];
    1132: vector<TYPE> dijkstra(int start){
    1133:
                                                      vector<TYPE> dist(MAX N, INFF);
    1134:
                                                      dist[start] = 0;//dist[i] := \tilde{a} \ 200 \ 200 start -> i \tilde{a} \ 201 \tilde{a} \ 201 \tilde{a} \ 201 \tilde{a} \ 201 \tilde{a}
 \237-è•\235é\233¢
    1135:
                                                      priority_queue<pair<TYPE, int>, vector<pair<TYPE, int> >, greater<pair<TYPE,</pre>
   int> > > que;
    1136:
                                                      que.push(make_pair(0, start));
    1137:
                                                      while(!que.empty()){
    1138:
                                                                                 TYPE cost; int u;//ä»\212ã\201¼ã\201§ã\201«ã\201\213ã\201\213ã\201£ã
 1139:
                                                                                  cost = que.top().first, u = que.top().second;
    1140:
                                                                                  que.pop();
    1141:
                                                                                  if(dist[u] < cost) continue;</pre>
    1142:
                                                                                  for (auto tmp : G[u]){
    1143:
                                                                                                               int v = tmp.first; TYPE time = tmp.second;//é\232fæ\216¥ã
\202é\226\223
    1144:
                                                                                                               if(dist[v] > dist[u] + time) {//u->v}
    1145:
                                                                                                                                           dist[v] = dist[u] + time;
    1146:
                                                                                                                                           que.push(make_pair(dist[v], v));
    1147:
                                                                                  }
    1148:
    1149:
    1150:
                                                      return dist;
    1151: }
    1152:
    1153:
```

```
1154: /*
  1155: http://joi2016yo.contest.atcoder.jp/tasks/joi2016yo e
  1156: (http://joi2016yo.contest.atcoder.jp/submissions/1201083)
  1157: http://judge.u-aizu.ac.jp/onlinejudge/description.jsp?id=2402
  1158: */
  1159:
  1160:
  1162: ###### ./graph/dijkstra_.Restoration.cpp ########
  1164:
  1165: const int MAX N = 210;
  1166: vector<pair<int, int> > G[MAX N];
  1167: vector<int> dijkstra(int start, int goal){//ã\202¹ã\202;ã\203¼ã\203\210ã\201"ã\202´ã
1168:
                                                             vector<long long> dist(MAX N, INF);
  1169:
                                                             vector<int> pre(MAX_N, -1);//pre[i] := iã\201@å\211\215ã\201@é\202ç\2021
 1170:
                                            dist[start] = 0; //dist[i] := \tilde{a} 200 \times 200 \times tart -> i \tilde{a} \times 201 \tilde{a} \times 
\235é\233¢
  1171:
                                            priority queue<pair<int, int>, vector<pair<int, int> >, greater<pair<int, int> >
  > que;
  1172:
                                            que.push(make_pair(0, start));
  1173:
                                            while(!que.empty()){
  1174:
                                                             int cost, u, t;//ä»\212ã\201¼ã\201§ã\201\213ã\201\213ã\201\213ã\201\237æ
1175:
                                                             cost = que.top().first, u = que.top().second;
  1176:
                                                             que.pop();
  1177:
                                                             if(dist[u] < cost) continue;</pre>
  1178:
                                                              for (auto tmp : G[u]){
  1179:
                                                                             int v = tmp.first, time = tmp.second;//\epsilon \langle 232fx \langle 216\tilde{x} \rangle \langle 201 \langle 231\tilde{a} \rangle \langle 202 \langle 213\tilde{e} \rangle \langle 2
\202¢\202¹ ã\201\235ã\201®é \202¢\202¹ã\201¾ã\201§è;\214ã\201\217æ\231\202é\226\223
  1180:
                                                                            if(dist[v] > dist[u] + time){//u->v}
  1181:
                                                                                              dist[v] = dist[u] + time;
  1182:
                                                                                              pre[v] = u;
  1183:
                                                                                               que.push(make_pair(dist[v], v));
  1184:
                                                                              else if(dist[v] == dist[u] + time){//e^{2}/236} \times 233.6 \times 206 \times 234 \times 2008 \times 217
  1185:
                                                                                              pre[v] = min(pre[v], u);
  1186:
  1187:
  1188:
  1189:
                                            //cu\214è• - 復å\205\203
  1190:
                                            vector<int> path;
  1191:
                                            int s = start, t = goal;
  1192:
                                                              for (; t != s; t = pre[t]) path.push_back(t);
  1193:
                                                             path.push back(s);
  1194:
                                                             return path; //start-> * -> goal startã\201\213ã\202\211goalã\201\8ã
(2010x)(234)(200c)(237-cu)(214e)^{-}(e)(236x)(233)(e)(206x)(234)(200a)(217)
 1195: }
  1196:
  1197: /*
  1198: http://yukicoder.me/problems/no/160
  1199: */
  1200:
  1202: ########### ./graph/dijkstra_ex.h ###############
  1204:
  1205: int dist[110][30]; // tyouten time := cost
  1206: using tup = tuple<int, int, int>;
  1207: int dijkstra(int start, int goal, int limit){
  1208:
                                            dist[start][0] = 0;
  1209:
                                            rep(i, 110)rep(j, 30) dist[i][j] = INF;
  1210:
                                            priority_queue<tup, vector<tup>, greater<tup>> que; // cost v time
  1211:
                                            que.push(make_tuple(0, start, 0));
  1212:
                                            while(!que.empty()) {
  1213:
                                                              int cost, u, time; tie(cost, u, time) = que.top(); que.pop();
```

```
1214:
                                          //cout << "cost u time " << cost << " " << u << " " << time << endl;
                                                                                                                                                                                                                                                                             1276:
   1215:
                                          if(u == goal) return cost;
                                                                                                                                                                                                                                                                             1277:
                                                                                                                                                                                                                                                                                                                     void add edge(int from, int to, int cap){
                                                                                                                                                                                                                                                                             1278:
                                                                                                                                                                                                                                                                                                                                           G[from].push_back((edge){to, cap, (int)G[to].size()});//from -> to
   1216:
                                          if(dist[u][time] < cost) continue;</pre>
                                                                                                                                                                                                                                                                             1279:
   1217:
                                          for(auto tmp : G[u]) {
                                                                                                                                                                                                                                                                                                                                           G[to].push_back((edge){from, 0, (int)G[from].size() - 1});//to -> fr
   1218:
                                                     int v, nc, nt; tie(v, nc, nt) = tmp;
   1219:
                                                      int ncost = cost + nc;
                                                                                                                                                                                                                                                                             1280:
   1220:
                                                      int ntime = time + nt;
                                                                                                                                                                                                                                                                             1281:
                                                                                                                                                                                                                                                                                                                     //å¢\227å\212\ \tilde{a}\203\221\tilde{a}\202^1\tilde{a}\202\222x\216¢\tilde{a}\201\231
   1221:
                                                      if(ntime > limit) continue;
                                                                                                                                                                                                                                                                             1282:
                                                                                                                                                                                                                                                                                                                     int dfs(int v, int t, int f){
   1222:
                                                      if(dist[v][ntime] > ncost) {
                                                                                                                                                                                                                                                                             1283:
                                                                                                                                                                                                                                                                                                                                           if(v == t) return f;
   1223:
                                                                 dist[v][ntime] = ncost;
                                                                                                                                                                                                                                                                             1284:
                                                                                                                                                                                                                                                                                                                                           used[v] = true;
   1224:
                                                                 que.push(make tuple(ncost, v, ntime));
                                                                                                                                                                                                                                                                             1285:
                                                                                                                                                                                                                                                                                                                                            for (int i = 0; i < G[v].size(); ++i){</pre>
   1225:
                                                                                                                                                                                                                                                                             1286:
                                                                                                                                                                                                                                                                                                                                                                  edge &e = G[v][i];
   1226:
                                                                                                                                                                                                                                                                             1287:
                                                                                                                                                                                                                                                                                                                                                                  if(!used[e.to] && e.cap > 0){
                                                                                                                                                                                                                                                                                                                                                                                         int d = dfs(e.to, t, min(f, e.cap));
   1227:
                                                                                                                                                                                                                                                                             1288:
                               return INF; // -1
                                                                                                                                                                                                                                                                             1289:
                                                                                                                                                                                                                                                                                                                                                                                         if(d > 0)
   1228:
   1229: }
                                                                                                                                                                                                                                                                             1290:
                                                                                                                                                                                                                                                                                                                                                                                                               e.cap -= d;
   1230:
                                                                                                                                                                                                                                                                             1291:
                                                                                                                                                                                                                                                                                                                                                                                                               G[e.to][e.rev].cap += d;
   1231: /*
                                                                                                                                                                                                                                                                             1292:
                                                                                                                                                                                                                                                                                                                                                                                                               return d;
  1232: aoj 2585
                                                                                                                                                                                                                                                                             1293:
   1233: */
                                                                                                                                                                                                                                                                             1294:
   1234:
                                                                                                                                                                                                                                                                             1295:
   1296:
                                                                                                                                                                                                                                                                                                                                           return 0;
   1236: ############## ./graph/euler tour.h ##############
                                                                                                                                                                                                                                                                             1297:
   1298:
                                                                                                                                                                                                                                                                                                                      //sã\201\213ã\202\211tã\201.ã\201@æ\234\200大æu\201
   1238:
                                                                                                                                                                                                                                                                             1299:
                                                                                                                                                                                                                                                                                                                     int max_flow(int s, int t){
                                                                                                                                                                                                                                                                                                                                           int flow = 0;
   1239: class Euler_Tour{
                                                                                                                                                                                                                                                                             1300:
   1240: public:
                                                                                                                                                                                                                                                                             1301:
                                                                                                                                                                                                                                                                                                                                           while(1){
   1241:
                                                                                                                                                                                                                                                                             1302:
                                                                                                                                                                                                                                                                                                                                                                  memset(used, 0, sizeof(used));
                                          vector<vector<int> > g;
   1242:
                                          //begin[v],end[v]ã\201~ã\201\235ã\202\214ã\201\236ã\202\214vã\201\214ã\202ªã
                                                                                                                                                                                                                                                                             1303:
                                                                                                                                                                                                                                                                                                                                                                  int f = dfs(s, t, INF);
\202¤ã\203@ã\203¼ã\203\204ã\202¢ã\203¼ä,\212ã\201§æ\234\200å\210\235ã\201"æ\234\200å\214ã
                                                                                                                                                                                                                                                                             1304:
                                                                                                                                                                                                                                                                                                                                                                  if(f == 0) return flow;
201 < 217 %202 214202 213202 223202 203207 203207 203207 203207 202207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 203207 2007207 207207 207207 207207 207207 207207 207207 207207 207207 207207 207207 
                                                                                                                                                                                                                                                                             1305:
                                                                                                                                                                                                                                                                                                                                                                  flow += f;
                                          //[begin[v], end[v])ã\201\214vã\202\222æ 1ã\201"ã\201\231ã\202\213é\203"å
  1243:
                                                                                                                                                                                                                                                                             1306:
\210\206æ\234"
                                                                                                                                                                                                                                                                             1307:
   1244:
                                          vector<int> euler_tour, begin, end;
                                                                                                                                                                                                                                                                             1308: };
   1245:
                                          Euler_Tour(int n) : g(n), begin(2 * n), end(2 * n){};//n\tilde{a}\\ 201^-\tilde{e} \\ 202\chi\\ 202\chi\\ 202\chi\\ 201^-\tilde{e} \\ 202\chi\\ 202\chi\\ 201^-\tilde{e} \\ 202\chi\\ 201^-\tilde{e} \\ 202\chi\\ 201^-\tilde{e} \\ 202\chi\\ 201^-\tilde{e} \\ 201^-\ti
                                                                                                                                                                                                                                                                             1309:
\225°
                                                                                                                                                                                                                                                                             1310:
   1246:
                                          int k = 0, root = 0;
   1247:
                                          void dfs(int curr, int par){//c\217\%\234\234\200\200parent
                                                                                                                                                                                                                                                                             1312: æ\234\200大æu\201 2é\203"ã\203\236ã\203\203ã\203\201ã\203³ã\202°
                                                                 begin[curr] = k;
   1248:
                                                                                                                                                                                                                                                                             1313: http://codeforces.com/contest/777/problem/B
   1249:
                                                                 euler_tour.push_back(curr);
   1250:
                                                                                                                                                                                                                                                                             1315:
                                                                 k++;
   1251:
                                                                 for(auto next : q[curr]){
                                                                                                                                                                                                                                                                             1252:
                                                                                       if(next == par) continue;
                                                                                                                                                                                                                                                                             1253:
                                                                                       dfs(next, curr);
   1254:
                                                                                       euler_tour.push_back(curr);
                                                                                                                                                                                                                                                                             1319:
   1255:
                                                                                      k++;
                                                                                                                                                                                                                                                                             1320: class Tree_lca{
   1256:
                                                                                                                                                                                                                                                                             1321: public:
   1257:
                                                                 end[curr] = k;
                                                                                                                                                                                                                                                                             1322:
                                                                                                                                                                                                                                                                                                                     static const int MAXLOG V = 25;
   1258:
                                                                                                                                                                                                                                                                             1323:
                                                                                                                                                                                                                                                                                                                     vector<vector<int> > G;
   1259: };
                                                                                                                                                                                                                                                                             1324:
                                                                                                                                                                                                                                                                                                                     int V, root;//é \202ç\2021æ\225° æ 1
   1260:
                                                                                                                                                                                                                                                                            1325:
                                                                                                                                                                                                                                                                                                                      // parent[k][v] := é \202c\202\vã\201\213ã\202\2112^kå\233\236è|aå\201´ã\201
                                                                                                                                                                                                                                                                            \langle c \rangle 231 \rangle \tilde{a} \langle 201f\tilde{a} \rangle 201/f\tilde{a} \langle 210\% \rangle 201/224\tilde{a} \langle 201 \rangle 231\tilde{a} \langle 202 \rangle 213\tilde{e} \langle 202c \rangle 202^1 \\ \langle e \rangle 200 \rangle 232\tilde{a} \langle 202 \rangle 212\tilde{e} \langle 201 \rangle 231\tilde{e} \langle 202c \rangle 202^2 \\ \langle e \rangle 200 \rangle 232\tilde{a} \langle 202 \rangle 212\tilde{e} \langle 201 \rangle 231\tilde{e} \langle 202c \rangle 212\tilde{e} \langle 202c \rangle 2
   1261: /*
   1262: http://codeforces.com/contest/383/problem/C
                                                                                                                                                                                                                                                                           \216\\alpha\\201\\216\alpha\\202\\213\alpha\\201\"-1)
   1263: */
                                                                                                                                                                                                                                                                            1326:
                                                                                                                                                                                                                                                                                                                     vector<int> parent[MAXLOG_V];
                                                                                                                                                                                                                                                                             1327:
                                                                                                                                                                                                                                                                                                                     vector<int> depth;//x ^1\tilde{a}\201\213\tilde{a}\202\211\tilde{a}\201@x•±\tilde{a}\201\225
   1264:
                                                                                                                                                                                                                                                                             1328:
   1329:
   Tree_lca(int V, int root) : V(V), root(root){
                                                                                                                                                                                                                                                                             1330:
   G.resize(V);
                                                                                                                                                                                                                                                                             1331:
   1268:
                                                                                                                                                                                                                                                                                                                                            for (int i = 0; i < MAXLOG_V; ++i) parent[i].resize(V);</pre>
   1269: const int MAX_V = 2010; //å¿\205è/\201ã\201ªé\202ç\202¹æ\225°
                                                                                                                                                                                                                                                                             1332:
                                                                                                                                                                                                                                                                                                                                           depth.resize(V);
   1270: struct Flow{
                                                                                                                                                                                                                                                                             1333:
   1271:
                                                                                                                                                                                                                                                                             1334:
                                                                                                                                                                                                                                                                                                                     void unite(int u, int v){ //u-vã\202\222unite
                                          struct edge{
   1272:
                                                                 int to, cap, rev;
                                                                                                                                                                                                                                                                             1335:
                                                                                                                                                                                                                                                                                                                                           G[u].push_back(v);
   1273:
                                                                                                                                                                                                                                                                             1336:
                                                                                                                                                                                                                                                                                                                                           G[v].push_back(u);
   1274:
                                          vector<edge> G[MAX_V];//é\232£æ\216¥ã\203ªã\202¹ã\203\210
                                                                                                                                                                                                                                                                             1337:
   1275:
                                          bool used[MAX V];
                                                                                                                                                                                                                                                                             1338:
                                                                                                                                                                                                                                                                                                                     int dist(int u, int v){ // u-v ã\201@è•\235é\233¢
```

Page 13 of 17

1339: int p = lca(u, v); 1340: return (depth[u] - depth[p]) + (depth[v] - depth[p]); 1341: 1342: void init(){ //parent[0]\(\tilde{a}\) 201" depth\(\tilde{a}\) 202\\(222\) \(\tilde{a}\) 210\\\(235\) \(234\) \(237\) \(\tilde{a}\) \(214\) \(226\) 1343: dfs(root, -1, 0); 1344: //parentã\202\222å\210\235æ\234\237å\214\226 2^kã\201\224ã\201"ã\201 $@\tilde{a} \ 203 \ 206 \tilde{a} \ 203 \ 226 \tilde{a} \ 203 \ \tilde{a} \ 202 \ 222 \tilde{a} \ 120$ 1345: for (int k = 0; k + 1 < MAXLOG V; ++k) { 1346: for (int v = 0; v < V; ++v)1347: if(parent[k][v] < 0) parent[k + 1][v] = -1;//root(ro otã\202\210ã\202\212ä,\212)ã\201\214è|a else parent[k + 1][v] = parent[k][parent[k][v]];//ã \202\213 1349: 1350: 1351: 1352: private: 1353: //ï¼\221ã\201¤ä,\212ã\201®è|ªã\201"敱ã\201\225ã\202\222è"-å®\232 1354: void dfs(int v, int p, int d){//ã\203\216ã\203\4ã\203\211c\225ªå\217•ã\200 $(200e)^{3}(2010e)^{3}(203(2166203)^{2}(203(211e)^{225})^{3}(217 \cdot \tilde{a}(200(200a) + \tilde{a}(201(225))^{2})^{2}(2166200)^{2}(2162000)^{2}(2166200)^{2}(2166200)^{2}(2166200)^{2}(2166200)^{2}(2166200)^{2}(2166200)^{2}(2166200)^{2}(2166200)^{2}(2166200)^{2}(2166200)^{2}(2166200)^{2}(2166200)^{2}(2166200)^{2}(2166200)^{2}(2166200)^{2}(2166200)^{2}(2166200)^{2}(2166200)^{2}(2166200)^{2}($ 1355: parent[0][v] = p; depth[v] = d;1356: 1357: for(auto u : G[v]){ 1358: if(u != p) dfs(u, v, d + 1);1359: 1360: 1361: //u, vã\201@LCAã\202\222æ±\202ã\202\201ã\202\213 int lca(int u, int v){ 1362: 1363: if (depth[u] > depth[v]) swap(u, v); 1364: for (int k = 0; k + 1 < MAXLOG_V; k++) $\{ /u\tilde{a} \geq 201 \geq 14v\tilde{a} \geq 201 \cdot \tilde{a} \geq 220 \geq 214 \}$ 1365: if (((depth[v] - depth[u]) >> k) & 1) { 1366: $v = parent[k][v]; //bit\tilde{a}\langle 201\langle 214c \langle \langle 213\tilde{a}\langle 201\tilde{a}\tilde{a}\rangle 201|\tilde{a}$ \201\204\alpha\202\214\alpha\201\\alpha\200\201\end{a}\alpha\2222^ke\delta\alpha\201\alpha\201\204\alpha\201\217 1367: 1368: 1369: if (u == v) return u; $//2å\210\206x\216$ ¢¢ $^{\circ}$ ¢ã\201§LCAã\202\222x±\202ã\202\201ã\202\213ã\200 \202\213 1371: for (int k = MAXLOG V - 1; k >= 0; k--) { if (parent[k][u] != parent[k][v]) {//LCAã\201~ã\201\237ã\201@ $\tilde{a} \setminus 202 \setminus 212\varsigma \setminus 235 \setminus 200\tilde{a} \setminus 201 \setminus 217\acute{e} \setminus 231 \setminus 220\varsigma \setminus 225 \setminus 214\tilde{a} \setminus 201 \\ / \tilde{a} \setminus 201 \\ / \tilde{a}$ \207\(\tilde{a}\)\201\233\(\tilde{a}\)\201\204\(\tilde{a}\)\201\204 1373: u = parent[k][u]; 1374: v = parent[k][v]; 1375: 1376: 1377: return parent[0][u];//1ã\201¤ä,\212ã\201\214LCA 1378: 1379: }; 1380: 1381: /* 1382: init()ã\202\222å¿\230ã\202\214ã\201ªã\201\204ã\202\210ã\201\206ã\201« 1383: */ 1384: 1388: 1389: ///引é\200£cu\220æ\210\220å\210\206å\210\206è§£ (Kosaraju)/// 1390: #define MAX_V 10000//é \202ç\2021æ\225° 1391: 1392: int V; 1393: vector<int> G[MAX V], rG[MAX V];

```
1394: vector<int> vs;
  1395: bool used[MAX V];
  1396: /(cmp[v] = cmp[U]\tilde{a} \langle 201^a\tilde{a} \langle 202 \langle 211\tilde{a} \langle 200 \rangle 201\acute{e} \langle 202c \langle 202^1u, v\tilde{a} \langle 201^-\mathring{a} \langle 220 \rangle 214\tilde{a} \langle 201 \rangle 230\mathring{a} 
é\200£cu\220æ\210\220å\210\206
  1397: //cmpã\201«å\205¥ã\201fã\201\ã\201\204ã\202\213c\225ªå\217•ã\201-é\200fcu\220ã\201
\227ã\201\ã\201\204ã\202\213ã\202\ã\203\@ã\203\225ã\201\224ã\201\ã\201\æã\200\201ā\2003\210ã
ã\200\201ã\203«ã\203¼ã\203\227é\203″å\210\206ã\201<sup>-</sup>å\220\214ã\201\230ç\225ªå\217•ã\201«ã\201
aã\201£ã\201¦ã\201\204ã\202\213ã\200\202
 1398: int cmp[MAX V];//cmp[v] := é\202c\202\vã\201\214å\220«ã\201\%ã\202\214ã\202\213é\200
£çu\220æ\210\220å\210\206ã\201\214ã\201©ã\202\214ã\201ªã\201®ã\201\213ã\202\222礰ã\201\231ç
\225ªå\217•
  1399: //é\232£æ\216¥ã\203ªã\202¹ã\203\210ã\202\222ä\234ã\202\213
  1400: void add edge(int from, int to){//oorigin
                                            G[from].push_back(to);//ä, 216ã 201 210ã 202 211ã 202 214ã 201 237æ 234 211å
\220\221\aig{202\aig}\203\aig{203\225\aig}\201\bigs(232\xi\216\xi\203\aig)\aig\202\aig\203\210
                                            rG[to].push back(from);//ä_\216ã\201\210ã\202\211ã\202\214ã\201\237ã\202\23
\203@\\alpha\225\alpha\201\\\alpha\237\ckrela\215\alpha\202\\222\ella\200\\206\alpha\201\\227\alpha\201\\237\ckrela\211\alpha\220\\221\alpha\200\\201\\237\ckrela\211\alpha\220\\221\alpha\200\\201\\237\ckrela\201\\237\ckrela\234\\211\alpha\220\\221\alpha\200\\201\\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\\237\ckrela\201\237\ckrela\237\ckrela\2
ã\203©ã\203\225ã\201®é\232£æ\216¥ã\203ªã\202¹ã\203\210
  1403: }
  1404: //ä,\200å°/ç\233®ã\201®dfs
  1405: void dfs(int v){
  1406:
                                            used[v] = true;
  1407:
                                            for (int i = 0; i < G[v].size(); ++i){</pre>
  1408:
                                                                   if(!used[G[v][i]]) dfs(G[v][i]);
  1409:
  1410:
                                            vs.push back(v);//ã\201\223ã\202\214以ä,\212é\200°ã\202\201ã\201°ã\201\217ã
\201ª\(\alpha\) 201\(\alpha\) 237\(\alpha\) 202\(\alpha\) 201\(\alpha\) 213\(\alpha\) 201\(\alpha\) 
1411: }
  1412: //2å°/ç\233@ã\201@dfs
  1413: void rdfs(int v, int k){
  1414:
                                            used[v] = true;
  1415:
                                            cmp[v] = k; //é \ 202c \ 202^{1}v\tilde{a} \ 201 \ a^{3}\tilde{a} \ 201 \ 227\tilde{a} \ 201 \ \tilde{a} \ 200 \ 201kc \ 225^{a}c \ 233\tilde{a}
\201"å\daue \equiv \200\xi\chi\200\xi\200\\201\\201\xi\201\\202\\213\\201\\223\\201\\205\xi\201\\202\\213\\201\\203\\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\\205\xi\201\205\xi\201\\205\xi\201\200\xi\201\200\xi\201\200\xi\201\200\xi\20
ã\202\213
 1416:
                                             for (int i = 0; i < rG[v].size(); ++i){</pre>
  1417:
                                                                    if(!used[rG[v][i]]) rdfs(rG[v][i], k);
  1418:
  1419: }
  1420: int scc(){
  1421:
                                            \235æ\234\237å\214\226
 1422:
                                            vs.clear();//å\210\235æ\234\237å\214\226
  1423:
                                            for (int v = 0; v < V; ++v)
  1424:
                                                                    if(!used[v]) dfs(v);
  1425:
  1426:
                                            memset(used, 0 , sizeof(used));
  1427:
                                            int k = 0;//å%•é\200fçµ\220æ\210\220å\210\206ã\202\222å\210\206ã\201\221ã
\202\213ç\225ªå\217•
  1428:
                                             for (int i = vs.size() - 1; i >= 0; --i){//vs\tilde{a}/201 < \tilde{a}/205 \times \tilde{a}/201 \leq \tilde{a}/201}
1429:
                                                                    if(!used[vs[i]]){
  1430:
                                                                                            rdfs(vs[i], k); k++;
  1431:
  1432:
  1433:
                                            return k;//å%·é\200£cu\220æ\210\220å\210\206ã\201@æ\225°
  1434: }
  1435:
  1436:
  1438: ########### ./graph/scc and twosat.h ###########
  1440:
  1441: class strongly_connected_components{
  1442: public:
```

Page 14 of 17

```
1443:
                     int group cnt; // scca\201@æ\225°
                                                                                                                                    \203\225ã\201
 1444:
                                                                                                                                                                    add edge(a + (apos ? V : 0), b + (bpos ? 0 : V)); // not a -> b
                     vector<vector<int> > G, rG;
                                                                                                                                     1503:
                                                                                                                                     1504:
 1445:
                     vector<int> used, vs;
                                                                                                                                                                    add_edge(b + (bpos ? V : 0), a + (apos ? 0 : V)); // not b -> a
                                                                                                                                     1505:
 1446:
                     1506:
                                                                                                                                                        bool operator[](int k){
c¤°ã\201\231c\225ªå\217.
                                                                                                                                     1507:
                                                                                                                                                                    return res[k];
                    strongly_connected_components(const vector<vector<int> > &g, const vector<ve
                                                                                                                                     1508:
ctor<int> > &rq, int n):
                                                                                                                                     1509: };
 1448:
                                G(g), rG(rg), cmp(2 * n), used(2 * n)
                                                                                                                                     1510:
 1449:
                                //mainã\201@å\207/c\220\206
                                                                                                                                     1511: int main(void){
                                fill(used.begin(), used.end(), 0);
                                                                                                                                     1512:
 1450:
 1451:
                                for (int i = 0; i < G.size(); ++i){</pre>
                                                                                                                                     1513:
                                                                                                                                                         twosatisfiability sat(n); /* é \202c\2021ã\201@æ\225° */
 1452:
                                           if(!used[i]) dfs(i);
                                                                                                                                     1514:
                                                                                                                                                         /* ã\201\235ã\201@ã\201¼ã\201¼ã\202\222true, åp\211æ\233´ã\201\231ã\202\213ã
 1453:
                                                                                                                                    \201@ã\202\222falseã\201"ã\201\227ã\201|
                                fill(used.begin(), used.end(), 0);
 1454:
                                                                                                                                                                    x\235;ä»¶ã\202\222x°\200ã\201\237ã\201\225ã\201ªã\201\204ã\202\210ã
 1455:
                                int k = 0;
                                                                                                                                    1456:
                                for (int i = vs.size() - 1; i >= 0; --i){
                                                                                                                                    e%°ã\202\222e;%å\212 ã\201\231ã\202\213
                                                                                                                                                                    ex) true trueã\201\$\alpha\235;\delta\gamma\delta\202\222\alpha\201\237\delta\201\225\201\delta
 1457:
                                           if(!used[vs[i]]) rdfs(vs[i], k++);
                                                                                                                                     1516:
 1458:
                                                                                                                                    \201\204ã\201ªã\202\211
 1459:
                                group_cnt = k;
                                                                                                                                     1517:
                                                                                                                                                                    sat.add(i, false, j, false)
 1460:
                                                                                                                                     1518:
                     int operator[](int i){//é\200£çu\220æ\210\220å\210\206ã\201®ç\225ªå\217•ã
 1461:
                                                                                                                                     1519:
                                                                                                                                                        return 0;
\202\222e;\224ã\201\231
                                                                                                                                     1520: }
 1462:
                                return cmp[i];
                                                                                                                                     1521:
                                                                                                                                     1522: /*
 1463:
 1464: private:
                                                                                                                                     1523: http://yukicoder.me/problems/no/470
 1465:
                     void dfs(int curr){
                                                                                                                                     1524: http://yukicoder.me/problems/no/274
 1466:
                                used[curr] = true;
                                                                                                                                     1525: http://yukicoder.me/problems/no/483
 1467:
                                for(auto next : G[curr]){
                                                                                                                                     1526: http://codeforces.com/contest/776/problem/D
 1468:
                                          if(!used[next]) dfs(next);
                                                                                                                                     1527: */
 1469:
                                                                                                                                     1528:
                                                                                                                                     1470:
                                vs.push back(curr);
 1471:
                                                                                                                                     void rdfs(int curr, int k){
                                                                                                                                     1472:
 1473:
                                used[curr] = true;
                                                                                                                                     1532:
 1474:
                                cmp[curr] = k; //é \202c\202^{1}v\tilde{a}\201^{4}\201\227\tilde{a}\201/\tilde{a}\200\201kc\225
                                                                                                                                     1533: class strongly_connected_components{
<sup>a</sup>ç\233®ã\201"引é\200£çµ\220æ\210\220å\210\206ã\201\§ã\201\202ã\202\213ã\201\223ã\201"å\205¥ã
                                                                                                                                     1534: public:
\202\214ã\202\213
                                                                                                                                     1535:
                                                                                                                                                        vector<vector<int> > G, rG;
 1475:
                                for(auto next : rG[curr]){
                                                                                                                                     1536:
                                                                                                                                                        vector<int> used, vs;
 1476:
                                           if(!used[next]) rdfs(next, k);
                                                                                                                                                         1477:
                                                                                                                                    \202\213\eqrig \200\figu\220\alpha\210\220\alpha\210\206\alpha\201\214\alpha\201\@\alpha\202\214\alpha\201\alpha\alpha\201\alpha\alpha\201\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha
 1478:
                                                                                                                                    礰ã\201\231¢\225ªå\217•
 1479: };
                                                                                                                                     1538:
                                                                                                                                                        strongly_connected_components(int n): G(n), rG(n), cmp(n), used(n){}
                                                                                                                                     1539:
                                                                                                                                                         void add edge(int from, int to){
 1481: class twosatisfiability{
                                                                                                                                     1540:
                                                                                                                                                                    G[from].push back(to);
 1482: public:
                                                                                                                                     1541:
                                                                                                                                                                    rG[to].push_back(from);
 1483:
                     int V;
                                                                                                                                     1542:
 1484:
                                                                                                                                     1543:
                                                                                                                                                         int scc(){
                     vector<int> res; // 1:= 0:=
 1485:
                     vector<vector<int> > q, rq;
                                                                                                                                     1544:
                                                                                                                                                                    fill(used.begin(), used.end(), 0);
 1486:
                     two satisfiability (int n) : V(n), g(2 * n), rg(2 * n), res(n) {}
                                                                                                                                     1545:
                                                                                                                                                                    vs.clear();
 1487:
                                                                                                                                     1546:
                                                                                                                                                                    for (int i = 0; i < G.size(); ++i){</pre>
                     bool exec() {
                                                                                                                                                                               if(!used[i]) dfs(i);
 1488:
                                                                                                                                     1547:
 1489:
                                strongly_connected_components scc(g, rg, V);
                                                                                                                                     1548:
 1490:
                                                                                                                                     1549:
                                                                                                                                                                    fill(used.begin(), used.end(), 0);
                                for (int i = 0; i < V; i++) {</pre>
                                           if (scc[i] == scc[i + V]) return false;
 1491:
                                                                                                                                     1550:
 1492:
                                           res[i] = scc[i] > scc[i + V];
                                                                                                                                     1551:
                                                                                                                                                                    int k = 0;
 1493:
                                                                                                                                     1552:
                                                                                                                                                                    for (int i = vs.size() - 1; i >= 0; --i){
 1494:
                                                                                                                                     1553:
                                                                                                                                                                               if(!used[vs[i]]){
                                return true;
 1495:
                                                                                                                                     1554:
                                                                                                                                                                                          rdfs(vs[i], k++);
 1496:
                     void add_edge(int a, int b){
                                                                                                                                     1555:
 1497:
                                                                                                                                     1556:
                                g[a].push_back(b);
                                                                                                                                     1557:
 1498:
                                rg[b].push_back(a);
                                                                                                                                                                    return k; //scca\201@æ\225°
 1499:
                                                                                                                                     1558:
 1500:
                     //0~V-1: x_i
                                                                                                                                     1559: private:
 1501:
                     //V~2V-1: notx i
                                                                                                                                     1560:
                                                                                                                                                        void dfs(int curr){
 1502:
                     void add(int a, bool apos, int b, bool bpos){//a V b ã\202\222ã\202@ã\203@ã
                                                                                                                                     1561:
                                                                                                                                                                    used[curr] = true;
```

Page 15 of 17

```
1562:
                    for(auto next : G[curr]){
                                                                                      1626:
                                                                                                          if(k < 0 \mid \mid k > n) return 0;
1563:
                           if(!used[next]) dfs(next);
                                                                                      1627:
                                                                                                          11 ret = fac[n];
1564:
                                                                                      1628:
                                                                                                          (ret *= facInv[n - k]) %= MOD;
1565:
                    vs.push_back(curr);
                                                                                      1629:
                                                                                                          return ret;
1566:
                                                                                      1630:
1567:
             void rdfs(int curr, int k){
                                                                                      1631: private:
1568:
                    used[curr] = true;
                                                                                      1632:
                                                                                                  void inverse(void){
1569:
                    cmp[curr] = k; //é \202c\202^{1}v\tilde{a}\201^{4}\201\227\tilde{a}\201/\tilde{a}\200\201kc\225
                                                                                      1633:
                                                                                                          inv[1] = 1;
²ç\233®ã\201"引é\200£çµ\220æ\210\220å\210\206ã\201§ã\201\202ã\202\213ã\201\223ã\201"å\205¥ã
                                                                                      1634:
                                                                                                          for (int i = 2; i <= MAX_N; ++i){</pre>
                                                                                      1635:
                                                                                                                 // inv[i] = MOD - (MOD / i) * inv[MOD % i] % MOD;
\202\214ã\202\213
                    for(auto next : rG[curr]){
                                                                                      1636:
                                                                                                                 inv[i] = inv[MOD % i] * (MOD - MOD / i) % MOD;
1570:
1571:
                           if(!used[next]) rdfs(next, k);
                                                                                      1637:
1572:
                                                                                      1638:
1573:
                                                                                      1639:
                                                                                                  void factroial(void){
1574: };
                                                                                      1640:
                                                                                                          fac[0] = facInv[0] = 1;
1575:
                                                                                      1641:
                                                                                                          for (int i = 1; i <= MAX_N; ++i){</pre>
1642:
                                                                                                                 fac[i] = (fac[i - 1] * i) % MOD;
1577: ######### ./math/combination samll.cpp ##########
                                                                                      1643:
                                                                                                                 facInv[i] = (facInv[i - 1] * inv[i]) % MOD;
1644:
1579:
                                                                                      1645:
1580: const int MAX N = 60;
                                                                                      1646: };
1581: ll com[MAX_N][MAX_N];
                                                                                      1647:
1582: void combination(void) { //com[i][j] := iCj
                                                                                      1583:
             com[0][0] = 1;
                                                                                      1584:
             for (int i = 1; i <= MAX_N - 1; ++i){</pre>
                                                                                      1585:
                    for (int j = 0; j <= i; ++j) {
                                                                                      1651:
1586:
                           //ã\203\221ã\202¹ã\202«ã\203«ã\201®i¼\223è§\222å½¢
                                                                                      1652: bool isprime[20010];
1587:
                           if(j == 0) com[i][j] = com[i - 1][j];
                                                                                      1653: //ã\202~ã\203@ã\203\210ã\202¹ã\203\206ã\203\215ã\202¹
                                                                                      1654: void eratos(int m){
1588:
                            else com[i][j] = (com[i - 1][j] + com[i - 1][j - 1]);
1589:
                                                                                      1655:
                                                                                               for (int i = 0; i <= m; ++i) isprime[i] = true;</pre>
1590:
                                                                                      1656:
                                                                                               isprime[0] = isprime[1] = false;
1591: }
                                                                                      1657:
                                                                                               //iã\202\222æ\213ã\201\227ã\201\iã\201\8å\200\215æ\225°ã\202\222æ\210ã\201\227ã
1592:
                                                                                     \201/ã\201\204ã\201\217
1593: /*
                                                                                      1658:
                                                                                               for (int i = 2; i <= m; ++i){
1594: http://abc057.contest.atcoder.jp/tasks/abc057_d
                                                                                      1659:
                                                                                                  if(isprime[i]){
                                                                                      1660:
                                                                                                      for (int j = 2 * i; j <= m; j += i){
1595: (http://abc057.contest.atcoder.jp/submissions/1184866)
1596: */
                                                                                      1661:
                                                                                                          isprime[j] = false;
1597:
                                                                                      1662:
1663:
1599: ############ ./math/combinations.h ##############
                                                                                      1664:
1665: }
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:
                                                                                      1671: // x^k \pmod{m}
                                                                                      1672: long long powmod(long long x, long long k, long long m){
1607:
             MATH(){
1608:
                    inverse();
                                                                                      1673:
                                                                                                  long long ret = 1;
1609:
                    factroial();
                                                                                      1674:
                                                                                                   while(k){
1610:
                                                                                      1675:
                                                                                                          if(k & 1) ret = ret * x % m;
             11 \text{ nCk}(11 \text{ n, } 11 \text{ k}) \{ // n! / k!*(n-k)! \}
1611:
                                                                                      1676:
                                                                                                          x = x * x % m;
                                                                                                          k >>= 1;
1612:
                    if(k < 0 | | k > n) return 0;
                                                                                      1677:
                    ll ret = fac[n];
                                                                                      1678:
1613:
1614:
                    (ret *= facInv[k]) %= MOD;
                                                                                      1679:
                                                                                                  return ret;
                    (ret *= facInv[n - k]) %= MOD;
1615:
                                                                                      1680: }
1616:
                    return ret;
                                                                                      1681: // 1/a \mod(p(c^* x \backslash 225^\circ))
1617:
                                                                                      1682: long long invmod(long long a, long long p){
1618:
             ll nHk(ll n, ll k){// nHk} = n+k-1 C k = (n+k-1)! / k! * (n-1)!
                                                                                      1683:
                                                                                                  return powmod(a, p - 2, p);
1619:
                    if(n == 0 && k == 0) return 1;
                                                                                      1684: }
                    ll ret = fac[n + k - 1];
                                                                                      1685:
1620:
1621:
                    (ret *= facInv[k]) %= MOD;
                                                                                      1686: /*
1622:
                    (ret *= facInv[n - 1]) %= MOD;
                                                                                      1687: ã\203\225ã\202§ã\203«ã\203\236ã\203¼ã\201®å®\232ç\220\206ã\202\222ç\224"ã\201\204ã
1623:
                    return ret;
                                                                                     \201\, modã\201@\\200\206\a05\203
1624:
                                                                                      1688: http://tubo28.me/algorithm/modinv/
1625:
             11 \text{ nPk}(11 \text{ n, } 11 \text{ k}) \{ //nPk = n! / (n-k)! \}
                                                                                      1689:
```

```
1754:
                                                                                                                                                         if(k \% 2 == 0) return powmod(x * x % m, k / 2, m);
 1691: http://yukicoder.me/problems/no/492
                                                                                                                                      1755:
                                                                                                                                                         else return x * powmod(x, k - 1, m) % m;
 1692: (http://yukicoder.me/submissions/157384)
                                                                                                                                      1756: }
 1693: */
                                                                                                                                      1757:
                                                                                                                                      1758: /*
 1694:
 1759: http://arc044.contest.atcoder.ip/tasks/arc044 b
 1760: (http://arc044.contest.atcoder.jp/submissions/1151541)
 1761: */
                                                                                                                                      1762:
 1699: typedef vector<long long> vec;
                                                                                                                                      1763:
 1700: typedef vector<vec> mat;
                                                                                                                                      1701: mat mul(mat &A,mat &B) { // A * B ã\201@è"\210c@\227
                                                                                                                                      1765: ######### ./math/prime factorization.h #########
                    mat C(A.size(), vec(B[0].size()));
                                                                                                                                      1702:
 1703:
                     for(int i = 0; i < A.size(); i++){</pre>
                                                                                                                                      1767:
 1704:
                                for(int k = 0;k < B.size(); k++){</pre>
                                                                                                                                      1768: class primelib{
 1705:
                                           for(int i = 0; i < B[0].size(); i++){</pre>
                                                                                                                                      1769: public:
 1706:
                                                      C[i][j] = (C[i][j] + A[i][k] * B[k][j]) % (11)MOD; /
                                                                                                                                      1770:
                                                                                                                                                         vector<int> prime; //prime[i] := iã\201\214c´ æ\225°ã\201®å ´å\220\210ã\201«
                                                                                                                                     ã\201~0, 以å¤\226ã\201®ã\201~ã\201\215ã\201~iã\202\222å\211²ã\202\212ã\202\213æ\234\200大ã
/ mod
 1707:
                                                                                                                                     \201@c' æ\225°
 1708:
                                                                                                                                      1771:
                                                                                                                                                         vector<int> memo;
 1709:
                                                                                                                                      1772:
                                                                                                                                                         primelib(int size):prime(size + 10, 0), memo(size + 10, 0){
 1710:
                                                                                                                                      1773:
                    return C:
                                                                                                                                                                     prime_factorization(size + 1);
 1711: }
                                                                                                                                      1774:
 1712: mat pow(mat A, long long n){ // A^n ~a \ 201@e" \ 210c@ \ 227 O(m^3*logn)
                                                                                                                                      1775:
                                                                                                                                                          //[left,right]\(\tilde{a}\)\(201\)\(\tilde{a}\)\(201\)\(\tilde{a}\)\(201\)\(214\)\(\tilde{a}\)\(201\)\(\tilde{a}\)\(201\)\(\tilde{a}\)\(201\)\(\tilde{a}\)\(201\)\(\tilde{a}\)\(201\)\(\tilde{a}\)\(201\)\(\tilde{a}\)\(201\)\(\tilde{a}\)\(201\)\(\tilde{a}\)\(201\)\(\tilde{a}\)\(201\)\(\tilde{a}\)\(201\)\(\tilde{a}\)\(201\)\(\tilde{a}\)\(201\)\(\tilde{a}\)\(201\)\(\tilde{a}\)\(201\)\(\tilde{a}\)\(201\)\(\tilde{a}\)\(201\)\(\tilde{a}\)\(201\)\(\tilde{a}\)\(201\)\(\tilde{a}\)\(201\)\(\tilde{a}\)\(201\)\(\tilde{a}\)\(201\)\(\tilde{a}\)\(201\)\(\tilde{a}\)\(\tilde{a}\)\(201\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\tilde{a}\)\(\
 1713:
                    mat B(A.size(), vec(A.size()));
                                                                                                                                     ã\201\227ã\201|è; "ã\201\225ã\202\214ã\202\213
 1714:
                    for(int i = 0; i < A.size(); i++){</pre>
                                                                                                                                      1776:
                                                                                                                                                         void getlcm(int left, int right){
 1715:
                               B[i][i] = 1;
                                                                                                                                      1777:
                                                                                                                                                                     // fill(memo.beqin(), memo.end(), 0); //0ã\201§å\210\235æ\234\237å
 1716:
                                                                                                                                     \214\226
                     while(n > 0)
 1717:
                                                                                                                                      1778:
                                                                                                                                                                     for (int i = max(2, left); i <= right; ++i){</pre>
 1718:
                               if(n & 1) B = mul(B, A);
                                                                                                                                      1779:
                                                                                                                                                                                map<int, int> degree; //<x, k> := x^k
 1719:
                               A = mul(A, A);
                                                                                                                                      1780:
                                                                                                                                                                                int tmp = i;
 1720:
                               n >>= 1;
                                                                                                                                      1781:
                                                                                                                                                                                while(prime[tmp]){
 1721:
                                                                                                                                      1782:
                                                                                                                                                                                           degree[prime[tmp]]++;
 1722:
                     return B;
                                                                                                                                      1783:
                                                                                                                                                                                           tmp /= prime[tmp]; //x \ 234 \ 200 a \ \tilde{s} \ \tilde{a} \ 201 \ \tilde{g} \ \tilde{c} \ x \ 225 \ \tilde{a}
 1723: }
                                                                                                                                     \201§å\2112ã\202\213
 1724:
                                                                                                                                      1784:
 1725: int main(void){
                                                                                                                                      1785:
                                                                                                                                                                                degree[tmp]++; //æ@\213ã\201£ã\201\237ç´ æ\225°ã\202\222ã
 1726:
                    11 n; cin >> n;
                                                                                                                                     \202«ã\202/ã\203³ã\203\210
 1727:
                    mat A(2, vec(2));
                                                                                                                                                                                //memo[x] := [left, right] \tilde{a} 201 \tilde{a} 201 \tilde{a} 202 222 \tilde{c} \tilde{a} 233 
 1728:
                    A[0][0] = 100, A[0][1] = 1;
                                                                                                                                     1729:
                                                                                                                                      1787:
                    A[1][0] = 0, A[1][1] = 1;
                                                                                                                                                                                for(auto u : degree){
 1730:
                    auto ret = pow(A, n - 1);
                                                                                                                                      1788:
                                                                                                                                                                                           memo[u.first] = max(memo[u.first], u.second);
                                                                                                                                      1789:
 1731: }
                                                                                                                                      1790:
 1732:
 1733:
                                                                                                                                      1791:
                                                                                                                                      1792: private:
 1735: è;\214å\210\227æ\\224c@\227 O(m^3*logn)
                                                                                                                                      1793:
                                                                                                                                                          //prime[i] := i\tilde{a} 201 \ 214c^{\tilde{a}} \ 225^{\tilde{a}} \ 201^{\tilde{a}} \ \tilde{a} \ 220 \ 210\tilde{a} \ 201^{\tilde{a}} \ \tilde{a} \ 2201^{\tilde{a}}
                                                                                                                                     1736: A^n è;\214å\210\227ã\202\222è§£ã\201\217
 1737: http://yukicoder.me/problems/no/492
                                                                                                                                                         //ã\201\223ã\202\214ã\202\222ä½;ã\201\210ã\201\0(logn)ã\201\$ç´ å\233 æ\225°å
                                                                                                                                     \210\206è§£å\217-è\203%
 1738: (http://yukicoder.me/submissions/157402)
 1739:
                                                                                                                                      1795:
                                                                                                                                                         void prime factorization(int n){
                                                                                                                                      1796:
                                                                                                                                                          for (int i = 2; i <= n; i++) {</pre>
 1740: http://abc009.contest.atcoder.jp/tasks/abc009_4
 1741:
                                                                                                                                      1797:
                                                                                                                                                                     if (prime[i] == 0) {
 1742: \quad N*N\grave{e}_{1}\backslash 214\mathring{a}\backslash 210\backslash 227A\~{a}\backslash 202\backslash 222\~{a}/(234\~{a}\backslash 202\backslash 212\~{a}\backslash 201\backslash 237\~{a}\backslash 201\backslash 204\mathring{a} \quad \mathring{a}\backslash 220\backslash 210
                                                                                                                                      1798:
                                                                                                                                                                                for (int j = 2; j * i <= n; j++) {</pre>
                                                                                                                                                                                prime[i * j] = i;
 1743: mat \ A(N, \ vec(N));
                                                                                                                                      1799:
 1744:
                                                                                                                                      1800:
 1745: */
                                                                                                                                      1801:
 1746:
                                                                                                                                      1802:
 1803:
 1804: };
 1805:
 1750:
                                                                                                                                      1751: //x^k mod
                                                                                                                                      1807: ################# ./other/is ururu.h ###############
 1752: long long powmod(long long x, long long k, long long m) { // c^{1}\% (202 (212)) (224\% (201))
                                                                                                                                      1809:
 1753:
                    if(k == 0) return 1;
                                                                                                                                      1810: bool is ururu(int y){
```

```
1811:
            if(v % 4) return false;
1812:
            else if(y % 100) return true;
1813:
            else if(y % 400) return false;
 1814:
            else return true;
 1815: }
 1816:
 1818: ############### ./search/rollinghash.h ##############
 1820:
 1821: typedef unsigned long long ull;
 1822: const ull B = 100000007;
 1823: //aã\201<sup>-</sup>i½\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;</pre>
 1831:
         ull ah = 0, bh = 0;
 1832:
 1833:
         for (int i = 0; i < al; ++i) ah = ah * B + a[i];</pre>
 1834:
         for (int i = 0; i < al; ++i) bh = bh * B + b[i];</pre>
 1835:
 1836:
         for (int i = 0; i + al <= bl; ++i){</pre>
 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:
 1853: unsigned int xor128(){\frac{1}{4}}\pm 225\%
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;
            return (w = (w ^ (w >> 19)) ^ (t ^ (t >> 8)));
 1857:
1858: }
 1860: /*
 1861: http://yukicoder.me/problems/no/469
 1862: */
 1863:
```