[ICPC2009 国内予選問題 F] 締まっていこう

プログラム例

```
/* ACM-ICPC2009 国内予選 Problem F */
// http://www.waseda.jp/assoc-icpc2009/preliminary/contest/all_ja.html#section_F
// filename = pf.c
// コンパイル： cc -O2 pf.c
// 実行方法：   ./a.out < F0 > F0.result
// 確認方法：   ./check
#include <stdio.h>
#include <math.h>

int m;  // ひもの初期状態を表す頂点数（両端の穴を含む） 2 <= m <= 100
int n;  // 針の本数  0 <= n <= 100
struct point_t {
  int x;        // 0 <= x <= 1000
  int y;        // 0 <= y <= 1000
};
typedef struct point_t point;
typedef struct point_t vector;
struct state_t {
  int ptype;    // 0: 始点  1: 終点  2: 始点・終点以外のひもの点  3: 針
  point p;
  int turn;     // +1: left turn, -1: right turn
  double angle; // 角度
};
typedef struct state_t state;
point himo[100];        // ひもの初期状態
int hpt;
point hari[100];        // 針の位置
state path[100*100];    // パス
int plen;               // パスの長さ(配列 path に入っている点の数
state work[100];
int wpt;                // 配列 work に入っている点の数
double tlen;            // ひもの長さ

double pi;

// 起点 p1, 終点 p2 なるベクトルを返す
vector vec(point p1, point p2)
{
  vector v;
  v.x = p2.x - p1.x;
  v.y = p2.y - p1.y;
  return v;
}

// ベクトル v1 に対するベクトル v2 の向きが反時計回りなら 1, 時計回りなら -1
// v1 と v2 が平行なら 0
int rd(vector v1, vector v2)
{
  int op;
  op = v1.x*v2.y - v1.y*v2.x;
  if(op>0) return 1;
```

```
  else if(op<0) return -1;
  else return 0;
}

// ベクトル v1 とベクトル v2 がなす角を返す
// v1 に対して v2 が反時計回りなら生、時計回りなら負
double angle(vector v1, vector v2)
{
  double av1, av2, ip, acv;
  int t;
  ip = v1.x*v2.x + v1.y*v2.y;
  av1 = sqrt(v1.x*v1.x + v1.y*v1.y);
  av2 = sqrt(v2.x*v2.x + v2.y*v2.y);
  acv = acos(ip/(av1*av2));
  t = rd(v1, v2);
  if(t) acv *= t;
  return acv;
}

// 三角形 P1 p2 p3 の内部に P4 があれば 1 なければ 0
int is_inside(point p1, point p2, point p3, point p4)
{
  int d1, d2, d3, d;
  d1 = rd(vec(p1, p2), vec(p2, p4));
  d2 = rd(vec(p2, p3), vec(p3,p4));
  d3 = rd(vec(p3, p1), vec(p1,p4));
  d = d1+d2+d3;
  if(d == 3 || d == -3) return 1;
  else return 0;
}

void sort_work(int left, int right)
{
  int i;
  int lp, rp;
  double kv;
  state tmp;
  if(left >= right) return;
  kv = fabs(work[left].angle);
  lp  = left+1;
  while(lp <= right){
    if(kv == fabs(work[lp].angle)){
      lp++;
      continue;
    }
    if(fabs(work[lp].angle) > kv)
      kv = fabs(work[lp].angle);
    break;
  }
  if(lp > right) return;
  lp = left; rp = right;
  while(lp <= rp){
    while(kv <= fabs(work[rp].angle)) rp--;
    while(fabs(work[lp].angle) < kv) lp++;
```

```
    if(lp > rp) break;
    tmp = work[rp];
    work[rp] = work[lp];
    work[lp] = tmp;
    lp++;
    rp--;
  }
  sort_work(left, lp-1);
  sort_work(lp, right);
}

void release(int k)
{
  int i, j, turn, s1p;
  point p1, p2, p3, p4;
  vector v12, v23;
  state st, s1, s2, s3;
  s1p = k-1;
  s1 = path[k-1];
  s2 = path[k];
  s3 = path[k+1];
  p1 = s1.p;
  p2 = s2.p;
  p3 = s3.p;
  v12 = vec(p1,p2);
  v23 = vec(p2,p3);
  turn = rd(v12, v23);
  wpt = 0;
  for(i=0; i<n; i++){
    if(is_inside(p1, p2, p3, hari[i])){
      work[wpt].p = hari[i];
      work[wpt].angle = angle(v12, vec(p1,hari[i]));
      wpt++;
    }
  }
  sort_work(0, wpt-1);

  for(i=k; i<plen-1; i++)
    path[i] = path[i+1];
  plen--;

  for(i=0; i<wpt; i++){
    p4 = work[i].p;
    if(turn != rd(vec(path[k-1].p, p4), vec(p4,p3))){
      continue;
    }
    if(i==0){
      path[k-1].angle += angle(vec(p1,p2),vec(p1,p4));
    }
    for(j=plen-1; j>=k; j--)
      path[j+1] = path[j];
    plen++;
    st.ptype = 3;
    st.p = p4;
```

```
    st.turn = turn;
    st.angle = angle(vec(path[k-1].p, p4), vec(p4,p3));
    path[k++] = st;
    //    if(i==wpt-1){
    //      path[k].angle += angle(vec(p4,p3),vec(p2,p3));
    //    }
  }
  if(wpt > 0){
    path[k].angle += angle(vec(p4,p3),vec(p2,p3));
  }
  else {
    path[k-1].angle += angle(vec(p1,p2),vec(p1,p3));
    path[k].angle += angle(vec(p1,p3),vec(p2,p3));
  }
}

int main()
{
  int i,j,flag,trd, pno;
  double x1,y1,x2,y2;
  pi = acos(-1);                  // π
  pno = 1;
  while(1){
    scanf("%d %d", &m, &n);
    if(m==0 && n==0) break;
    //    fprintf(stderr,"%d: %d %d¥n", pno++,m, n);
    for(i=0; i<m; i++)
      scanf("%d %d", &himo[i].x, &himo[i].y);
    for(i=0; i<n; i++)
      scanf("%d %d", &hari[i].x, &hari[i].y);
    for(i=0; i<n*m; i++){
      path[i].ptype = -1;
      path[i].turn = 0;
      path[i].angle = 0;
    }
    tlen = wpt = plen = hpt = 0;

    path[0].ptype = 0;
    path[0].p = himo[0];
    path[0].turn = path[0].angle = 0;
    if(m==2) path[1].ptype = 1;
    else path[1].ptype = 2;
    path[1].p = himo[1];
    path[1].turn = path[1].angle = 0;
    plen = 2;
    for(i=2; i<m; i++){
      state st;
      if(i == m-1) st.ptype = 1;
      else st.ptype = 2;
      st.p = himo[i];
      st.turn = st.angle = 0;
      path[plen++] = st;
      release(plen-2);
```

```
      do {
        flag = 0;
        for(j=plen-2; j>=1; j--){
          trd = rd(vec(path[j-1].p, path[j].p), vec(path[j].p, path[j+1].p));
          if(trd == 0){
            if(path[j].turn > 0 && path[j].angle <0 && path[j].angle > -1.5*pi){
              release(j); flag = 1;
            }
            if(path[j].turn < 0 && path[j].angle > 0 && path[j].angle < 1.5*pi){
              release(j); flag = 1;
            }
          }
          else {
            if(fabs(path[j].angle) <= pi){
              if(path[j].turn != trd){
                release(j);
                flag = 1;
              }
            }
          }
        }
      } while(flag);
    }

    x1 = path[0].p.x;
    y1 = path[0].p.y;
    for(i=1; i<plen; i++){
      x2 = path[i].p.x;
      y2 = path[i].p.y;
      tlen += sqrt((x2-x1)*(x2-x1) + (y2-y1)*(y2-y1));
      x1 = x2;
      y1 = y2;
    }
    printf("%lf¥n",tlen);
  }
}
```

実行結果チェックプログラム

```
/* Result Check Program for ICPC2009 国内予選問題 F */
// filename check.c
// Complie:     cc -o check check.c -lm
// Execution:   ./check
#include <stdio.h>
#include <math.h>

FILE *fd1, *fd2;
double gosa = 0.001;

void compare()
{
  int line = 0;
  int ret1, ret2;
  double d1, d2;

  while(1){
    ret1 = fscanf(fd1, "%lf", &d1);
    ret2 = fscanf(fd2, "%lf", &d2);
    if((ret1 == EOF) && (ret2 == EOF)) break;
    if((ret1 == EOF) || (ret2 == EOF)){
      fprintf(stderr, "Length mismatch¥n");
      break;
    }
    line++;
    if(fabs(d1 - d2) >= gosa){
      printf("%d¥t%lf¥t%lf¥n", line, d1, d2);
    }
  }
}

int main()
{
  printf("F0¥n");
  fd1 = fopen("F0.ans","r");
  fd2 = fopen("F0.result", "r");
  compare();
  fclose(fd1);
  fclose(fd2);
  printf("F1¥n");
  fd1 = fopen("F1.ans","r");
  fd2 = fopen("F1.result", "r");
  compare();
  fclose(fd1);
  fclose(fd2);
  printf("F2¥n");
  fd1 = fopen("F2.ans","r");
  fd2 = fopen("F2.result", "r");
  compare();
  fclose(fd1);
  fclose(fd2);
  printf("F3¥n");
  fd1 = fopen("F3.ans","r");
```

```
  fd2 = fopen("F3.result", "r");
  compare();
  fclose(fd1);
  fclose(fd2);
  printf("F4¥n");
  fd1 = fopen("F4.ans","r");
  fd2 = fopen("F4.result", "r");
  compare();
  fclose(fd1);
  fclose(fd2);
}
```