A.

P

k-   .   *k*   ,

,   (*P*).

.

.

```
uses
    Math;

var
    i, n, j, k : longint;
    x, y, xt, yt : array [0..100] of extended;
    ans : extended;

begin
    reset(input, 'dual.in');
    rewrite(output, 'dual.out');

    read(n);
    for i := 0 to n - 1 do begin
        read(x[i], y[i]);
    end;
    read(k);

    for j := 1 to k do begin
        for i := 0 to n - 1 do begin
            xt[i] := (x[i] + x[(i + 1) mod n]) / 2;
            yt[i] := (y[i] + y[(i + 1) mod n]) / 2;
        end;
        x := xt;
        y := yt;
    end;

    ans := 0;
    for i := 0 to n - 1 do begin
        ans := ans + sqrt(sqr(x[(i + 1) mod n] - x[i]) +
sqr(y[(i + 1) mod n] - y[i]));
    end;

    write(ans);
end.
```

k   ,

O(kn),

n   k.

B. *K-*

, , 9! = 362880.

n

, k- .

.

```
procedure go(pos : integer);
var
  i : integer;
begin
  if (pos = n + 1) then begin
    for i := 1 to n - 1 do begin
      if abs(p[i] - p[i + 1]) > k then begin
        exit;
      end;
    end;
    inc(ans);
    exit;
  end;
  for i := 1 to n do begin
    if (not w[i]) then begin
      w[i] := true;
      p[pos] := i;
      go(pos + 1);
      w[i] := false;
    end;
  end;
end;
```

go(pos)

, pos p,

. , ,

, w, w[i] =

true i false . pos > n,

- , , k - .

, ans.

C.

.

,

, .

.

,

.

,   ,

2001,

.

field   (field:

array[1..2001, 1..2001] of integer),   .

,

.

field,

.

,   .

```
readln(s);
r := 1001;
c := 1001;
n := length(s);
for i := 1 to n do begin
   inc(field[r][c]);
   ch := s[i];
   if (ch = 'L') then begin
      c := c - 1;
   end else if (ch = 'R') then begin
      c := c + 1;
   end else if (ch = 'U') then begin
      r := r - 1;
   end else if (ch = 'D') then begin
      r := r + 1;
   end;
end;
inc(field[r][c]);
ans := 0;
for i := 1 to 2001 do begin
   for j := 1 to 2001 do begin
      if (field[i][j] > 1) then begin
         inc(ans);
      end;
   end;
end;
writeln(ans);
```

,   .

.

(,   –

).

(–

).

,   .

```
    readln(s);
    r := 0;
    c := 0;
    n := length(s);
    p := 1;
    for i := 1 to n do begin
       a[p][1] := r;
       a[p][2] := c;
       inc(p);
       ch := s[i];
       if (ch = 'L') then begin
          c := c - 1;
       end else if (ch = 'R') then begin
           c := c + 1;
       end else if (ch = 'U') then begin
           r := r - 1;
       end else if (ch = 'D') then begin
         r := r + 1;
       end;
    end;
    a[p][1] := r;
    a[p][2] := c;
    inc(p);

    for i := 1 to p do begin
       for j := i + 1 to p do begin
          if (a[j][1] < a[i][1]) or
             ((a[j][1] = a[i][1]) and (a[j][2] < a[i][2]))
             then begin
             t := a[i][1];
             a[i][1] := a[j][1];
             a[j][1] := t;
             t := a[i][2];
             a[i][2] := a[j][2];
             a[j][2] := t;
          end;
       end;
    end;
end;
```

a   ,

–   a[i][1]   ,

,   a[i][2] –   .   ,

,   .

,   .

, k ,

k ,

.

, —

, ,

.

.

```
    ans := 0;
    i := 1;
    while (i < p) do begin
        j := i;
        while (j < p) and (a[i][1] = a[j][1]) and (a[i][2] =
a[j][2]) do begin
            inc(j);
        end;
        if (j - i > 1) then begin
            inc(ans);
        end;
        i := j;
    end;
```

i ,

j .

i, ,

, , a[i].

, .

, , ,

ans, .

D.

- , n   10^5,

.

, , .

```
    read(x, y, z, w);

    read(n, k);

    a[1] := w;

    for i := 2 to n do begin

    a[i] := (x * a[i-1] + y) mod z;
```

```
    end;
```

a

,   .   ,

,

a[i] (*i* > 1),   *z.*   ,

a,   ,

0   *z*–1.   ,   w

10000,   ,   *a*   0

10000.

,   «   »

0   10000   ,   *a.*

```
for i := 1 to n do begin
inc(cnt[a[i]]);
end;
```

cnt[i]

cnt   *i*   *a.*

,   ,   0

(10000)

,   *k-*   .

b (0   *b*   10000).

b,   a   ,   *b,*

k.   .

.

$$\sum_{i=0}^{b} c\ \ [i] \le k, b \to m$$

,

$$10000^2 = 10^8$$.

, b $(b+1)$
. O(1).

, .

```
s := 0;
for b := 0 to 10000 do begin
    s := s + cnt[b];
    if (s >= k) then begin
        writeln (b);
        break;
    end;
end;
```

, ,
b. ,
b ,
break.